Online Learning for Big Data Analytics

Irwin King, Michael R. Lyu and Haiqin Yang
Department of Computer Science & Engineering
The Chinese University of Hong Kong

Tutorial presentation at IEEE Big Data, Santa Clara, CA, 2013

Outline

- Introduction (60 min.)
  - Big data and big data analytics (30 min.)
  - Online learning and its applications (30 min.)
- Online Learning Algorithms (60 min.)
  - Perceptron (10 min.)
  - Online non-sparse learning (10 min.)
  - Online sparse learning (20 min.)
  - Online unsupervised learning (20. min.)
- Discussions + Q & A (5 min.)

Outline

- Introduction (60 min.)
  - Big data and big data analytics (30 min.)
  - Online learning and its applications (30 min.)
- Online Learning Algorithms (60 min.)
  - Perceptron (10 min.)
  - Online non-sparse learning (10 min.)
  - Online sparse learning (20 min.)
  - Online unsupervised learning (20. min.)
- Discussions + Q & A (5 min.)

What is Big Data?

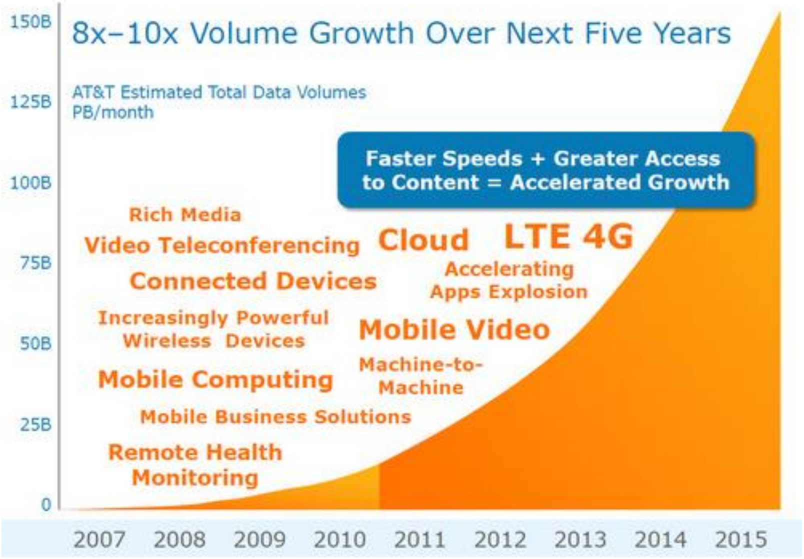
- There is not a consensus as to how to define Big Data

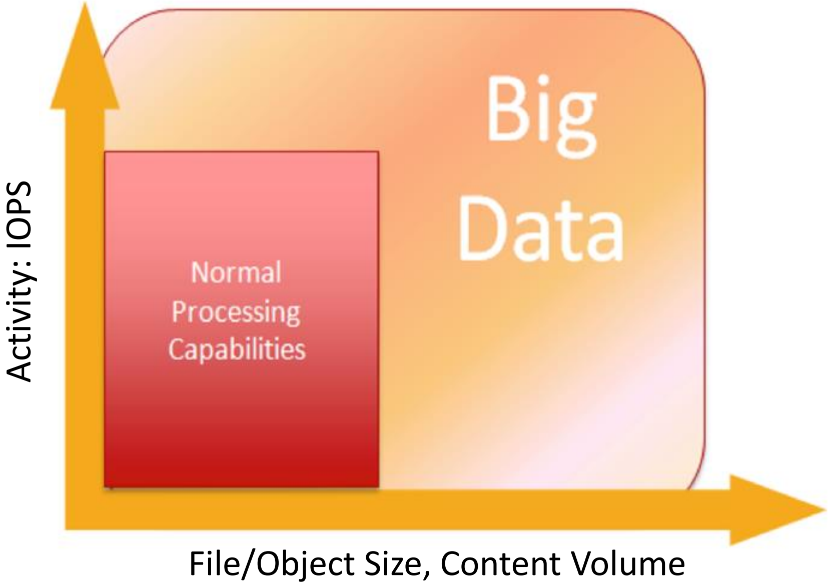
"A collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications." *- wiki*

"Big data exceeds the reach of commonly used hardware environments and software tools to capture, manage, and process it with in a tolerable elapsed time for its user population." *- Tera- data magazine article, 2011*

"Big data refers to data sets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze." *- The McKinsey Global Institute, 2011*

What is Big Data?





Big Data refers to datasets grow so large and complex that it is difficult to capture, store, manage, share, analyze and visualize within current computational architecture.

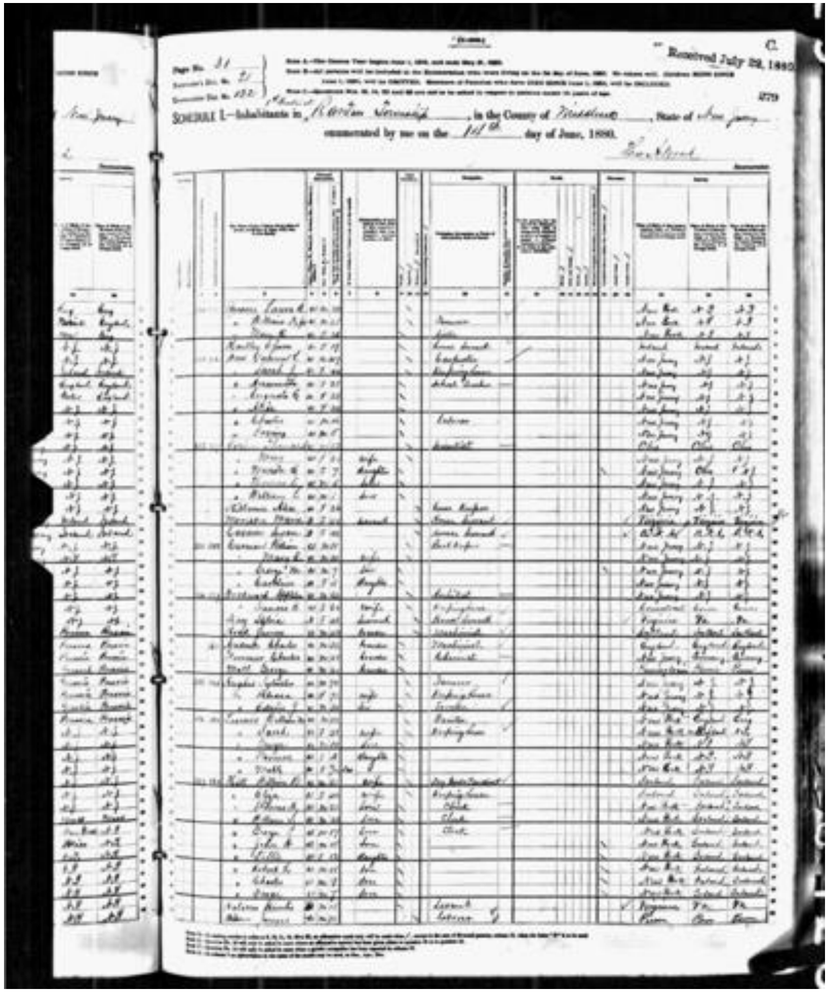
Evolution of Big Data

- Birth: 1880 US census
- Adolescence: Big Science
- Modern Era: Big Business

Birth: 1880 US census

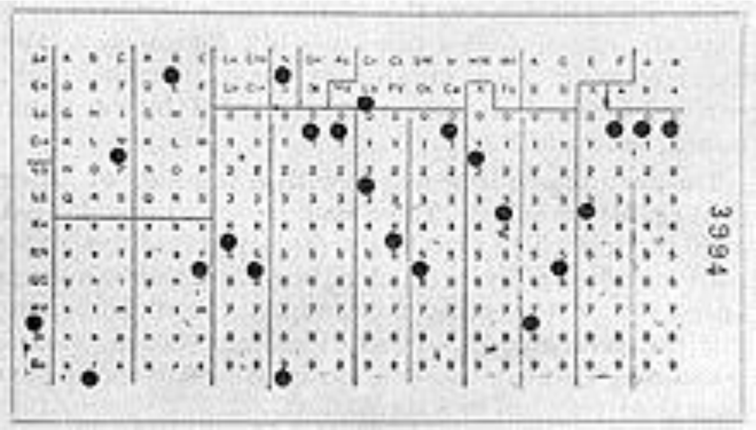
The First Big Data Challenge

- 1880 census
- 50 million people
- Age, gender (sex), occupation, education level, no. of insane people in household



The First Big Data Solution

- Hollerith Tabulating System
- Punched cards – 80 variables
- Used for 1890 census
- 6 weeks instead of 7+ years



Manhattan Project (1946 - 1949)

- $2 billion (approx. 26 billion in 2013)
- Catalyst for "Big Science"

Space Program (1960s)

- Began in late 1950s
- An active area of big data nowadays

Adolescence: Big Science

Big Science

- The International Geophysical Year
  - An international scientific project
  - Last from Jul. 1, 1957 to Dec. 31, 1958
- A synoptic collection of observational data on a global scale
- Implications
  - Big budgets, Big staffs, Big machines, Big laboratories

Summary of Big Science

- Laid foundation for ambitious projects
  - International Biological Program
  - Long Term Ecological Research Network
- Ended in 1974
- Many participants viewed it as a failure
- Nevertheless, it was a success
  - Transform the way of processing data
  - Realize original incentives
  - Provide a renewed legitimacy for synoptic data collection

Lessons from Big Science

- Spawn new big data projects
  - Weather prediction
  - Physics research (supercollider data analytics)
  - Astronomy images (planet detection)
  - Medical research (drug interaction)
  - …
- Businesses latched onto its techniques, methodologies, and objectives

Modern Era: Big Business

Big Science vs. Big Business

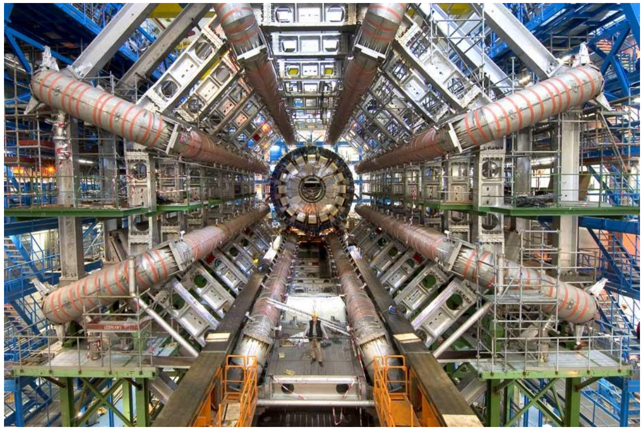
- Common
  - Need technologies to work with data
  - Use algorithms to mine data
- Big Science
  - Source: experiments and research conducted in controlled environments
  - Goals: to answer questions, or prove theories
- Big Business
  - Source: transactions in nature and little control
  - Goals: to discover new opportunities, measure efficiencies, uncover relationships

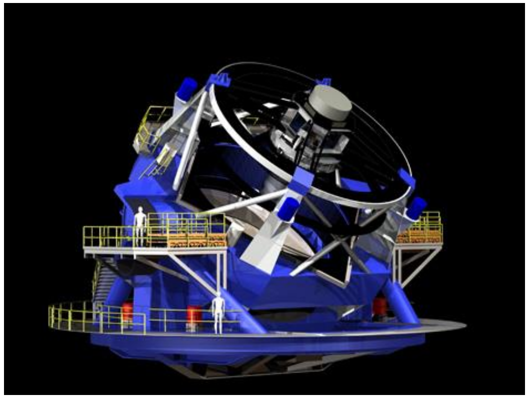
Big Data is Everywhere!

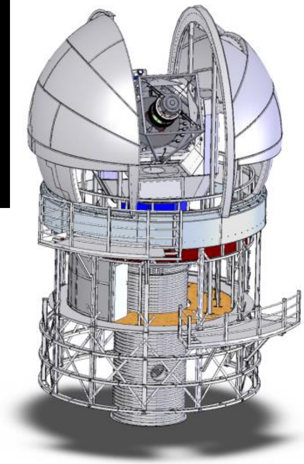
- Lots of data is being collected and warehoused
  - Science experiments
  - Web data, e-commerce
  - Purchases at department/ grocery stores
  - Bank/Credit Card transactions
  - Social Networks

Big Data in Science

- CERN - Large Hadron Collider
  - ~10 PB/year at start
  - ~1000 PB in ~10 years
  - 2500 physicists collaborating
- Large Synoptic Survey Telescope (NSF, DOE, and private donors)
  - ~5-10 PB/year at start in 2012
  - ~100 PB by 2025
- Pan-STARRS (Haleakala, Hawaii) US Air Force
  - now: 800 TB/year
  - soon: 4 PB/year







Big Data from Different Sources

12+ TBs
of tweet data
every day

? TBs of
data every
day

25+ TBs
of
log data
every day

30 billion RFID
tags today
(1.3B in 2005)

4.6
billion
camera
phones
world
wide

100s of
millions
of GPS
enabled
devices
sold
annually

76 million smart
meters in 2009…
200M by 2014

2+
billion
people
on the
Web by
end 2011

Big Data in Business Sectors

US health care

- $300 billion value per year
- ~0.7 percent annual productivity growth

Europe public sector administration

- £250 billion value per year
- ~0.5 percent annual productivity growth

Global personal location data

- $100 billion + revenue for service providers
- Up to $700 billion value to end users

US retail

- 60+ % increase in net margin possible
- 0.5-1.0 percent annual productivity growth

Manufacturing

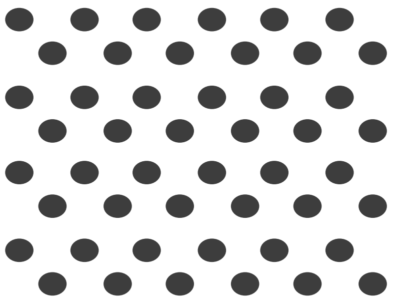
- Up to 50 percent decrease in product development, assembly costs
- Up to 7 percent reduction in working capital

SOURCE: McKinsey Global Institute

Characteristics of Big Data

- 4V: Volume, Velocity, Variety, Veracity

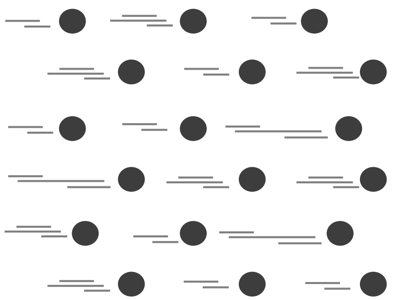
Volume



IDC
Analyze the Future

8 billion TB in 2015,
40 ZB in 2020
5.2TB per person

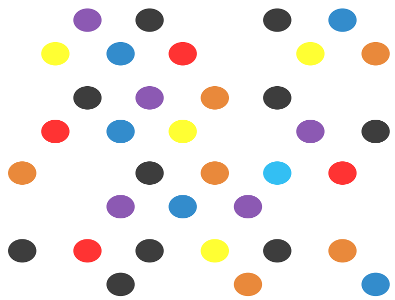
Velocity



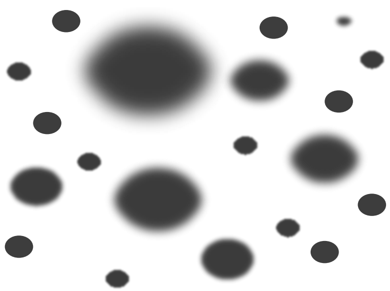
facebook

New sharing over 2.5 billion per day
new data over 500TB per day

Variety



Veracity



US predicts powerful quake this week

UAE body says it's a rumour;

By Wam/Agencies
Published Saturday, April 20, 2013

Big Data Analytics

- Definition: a process of inspecting, cleaning, transforming, and modeling big data with the goal of discovering useful information, suggesting conclusions, and supporting decision making
- Connection to data mining
  - Analytics include both data analysis (mining) and communication (guide decision making)
  - Analytics is not so much concerned with individual analyses or analysis steps, but with the entire methodology

Outline

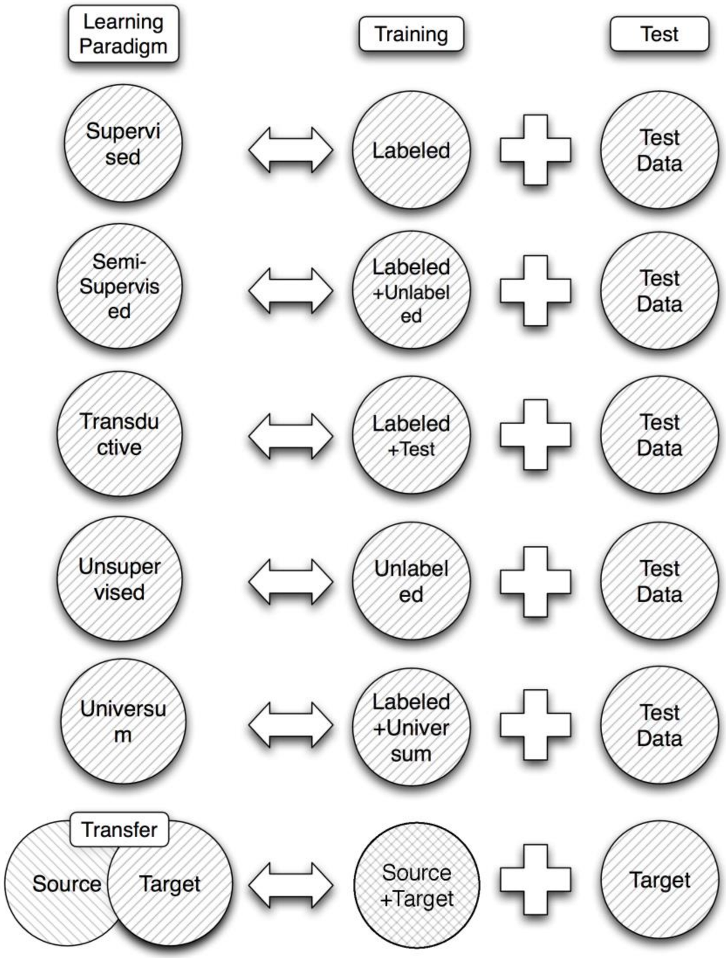
- Introduction (60 min.)
  - Big data and big data analytics (30 min.)
  - Online learning and its applications (30 min.)
- Online Learning Algorithms (60 min.)
  - Perceptron (10 min.)
  - Online non-sparse learning (10 min.)
  - Online sparse learning (20 min.)
  - Online unsupervised learning (20. min.)
- Discussions + Q & A (5 min.)

Challenges and Aims

- Challenges: capturing, storing, searching, sharing, analyzing and visualizing
- Big data is not just about size
  - Finds insights from complex, noisy, heterogeneous, longitudinal, and voluminous data
  - It aims to answer questions that were previously unanswered
- This tutorial focuses on **online learning techniques** for Big Data

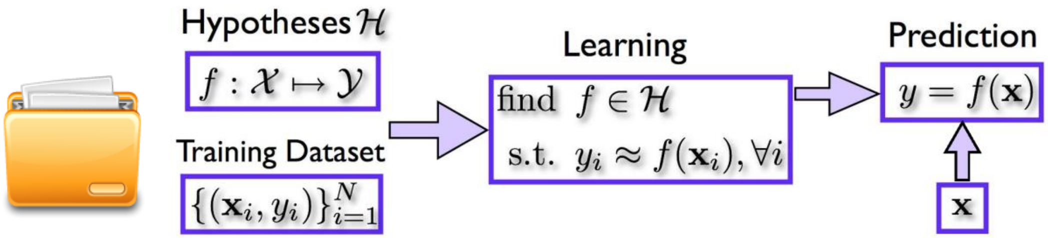
Learning Techniques Overview

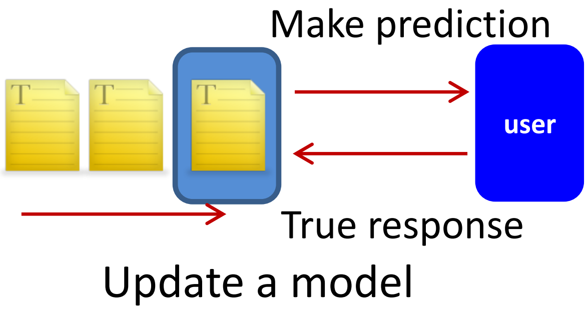
- Learning paradigms
  - **Supervised learning**
  - Semisupervised learning
  - Transductive learning
  - **Unsupervised learning**
  - Universum learning
  - Transfer learning



What is Online Learning?

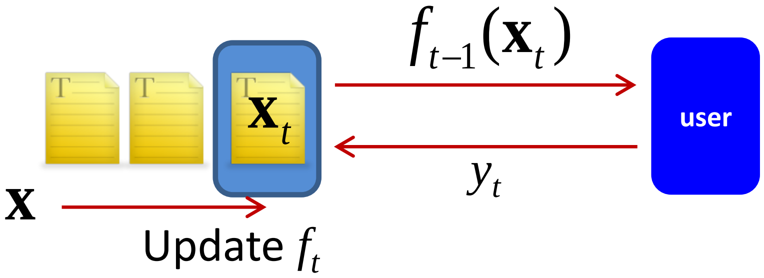
- Batch/Offline learning
  - Observe a **batch** of training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$
  - Learn a model from them
  - Predict new samples accurately
- Online learning
  - Observe a **sequence** of data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t)$
  - Learn a model **incrementally** as instances come
  - Make the sequence of online predictions accurately





Online Prediction Algorithm

- An initial prediction rule $f_0(\cdot)$
- For t=1, 2, …
  - We observe $\mathbf{x}_t$ and make a prediction $f_{t-1}(\mathbf{x}_t)$
  - We observe the true outcome y_t and then compute a loss $l(f(\mathbf{x}_t), y_t)$
  - The online algorithm updates the prediction rule using the new example and construct $f_t(\mathbf{x})$

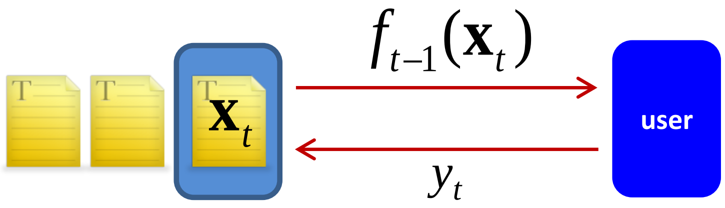


Online Prediction Algorithm

- The total error of the method is

$$\sum_{t=1}^{T} l(f_{t-1}(\mathbf{x}_t), y_t)$$

- Goal: this error to be as small as possible
- Predict unknown future one step a time: similar to generalization error



Regret Analysis

- $f_*(\cdot)$: optimal prediction function from a class H, e.g., the class of linear classifiers

$$f_*\left(\cdot\right) = \arg\min_{f \in H} \sum_{t=1}^{T} l(f(\mathbf{x}_t), y_t)$$

with minimum error after seeing all examples

- Regret for the online learning algorithm

$$\text{regret} = \frac{1}{T}\sum_{t=1}^{T}\left[l(f_{t-1}(\mathbf{x}_t), y_t) - l(f_*(\mathbf{x}_t), y_t)\right]$$

We want regret as small as possible

Why Low Regret?

- Regret for the online learning algorithm

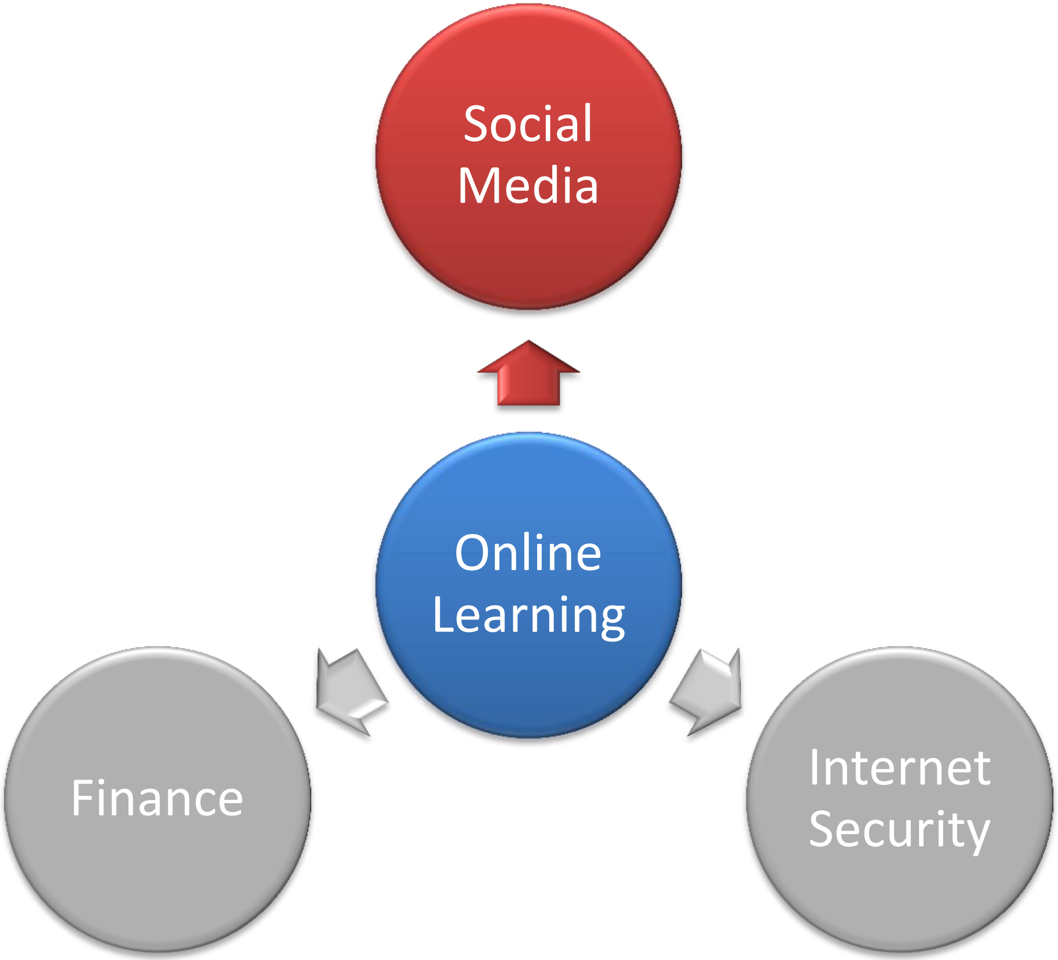
$$\text{regret} = \frac{1}{T}\sum_{t=1}^{T}\left[l(f_{t-1}(\mathbf{x}_t), y_t) - l(f_*(\mathbf{x}_t), y_t)\right]$$

- Advantages
  - We do not lose much from not knowing future events
  - We can perform almost as well as someone who observes the entire sequence and picks the best prediction strategy in hindsight
  - We can also compete with changing environment

Advantages of Online Learning

- Meet many applications for data arriving sequentially while predictions are required on-the-fly
  - Avoid re-training when adding new data
- Applicable in adversarial and competitive environment
- Strong adaptability to changing environment
- High efficiency and excellent scalability
- Simple to understand and easy to implement
- Easy to be parallelized
- Theoretical guarantees

Where to Apply Online Learning?



Social Media

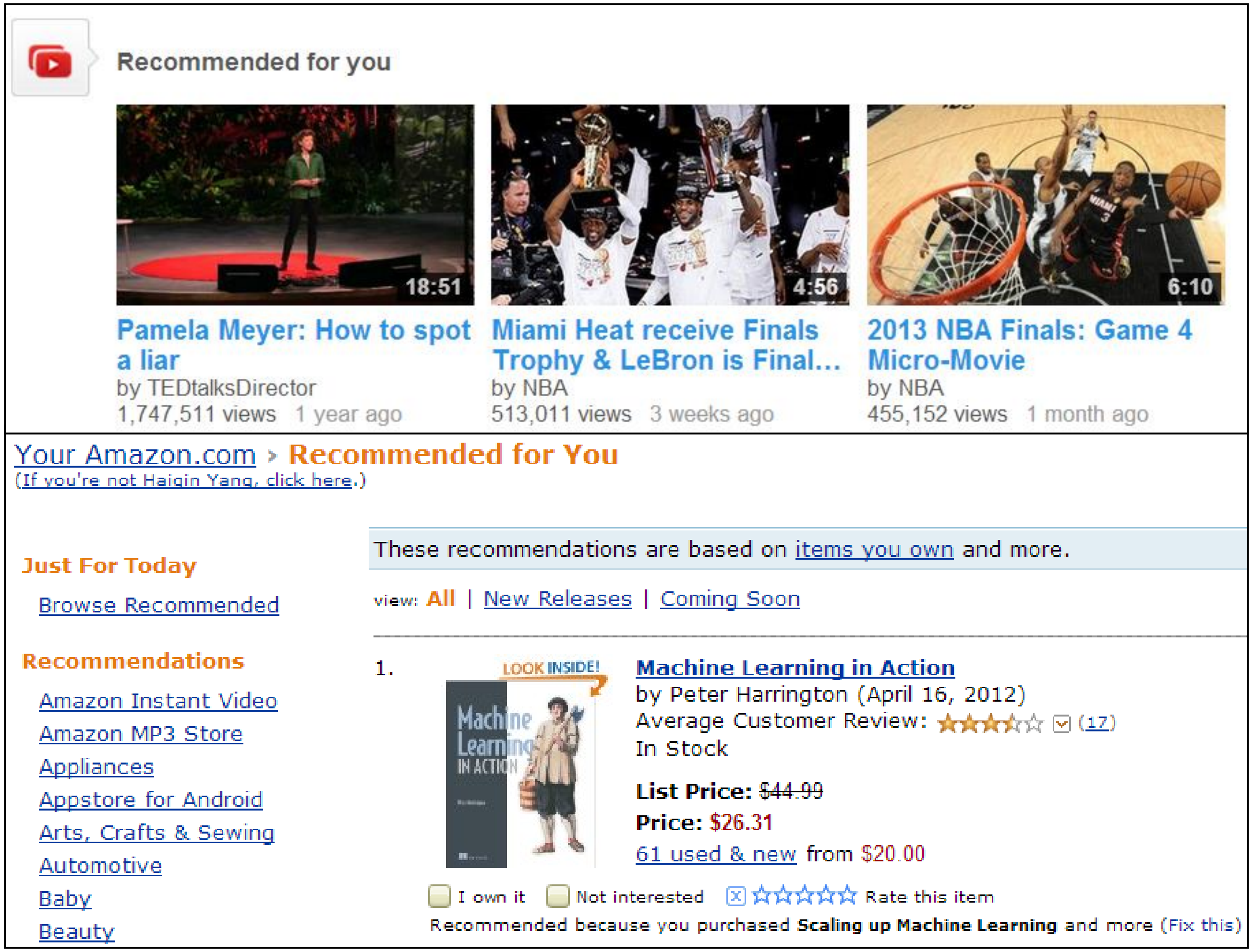
Online Learning

Finance

Internet Security

Online Learning for Social Media

- Recommendation, sentiment/emotion analysis



Recommended for you

18:51
Pamela Meyer: How to spot a liar
by TEDtalksDirector
1,747,511 views 1 year ago

4:56
Miami Heat receive Finals Trophy & LeBron is Final…
by NBA
513,011 views 3 weeks ago

6:10
2013 NBA Finals: Game 4 Micro-Movie
by NBA
455,152 views 1 month ago

Your Amazon.com › **Recommended for You**
(If you're not Haiqin Yang, click here.)

Just For Today

Browse Recommended

Recommendations

Amazon Instant Video
Amazon MP3 Store
Appliances
Appstore for Android
Arts, Crafts & Sewing
Automotive
Baby
Beauty

These recommendations are based on items you own and more.

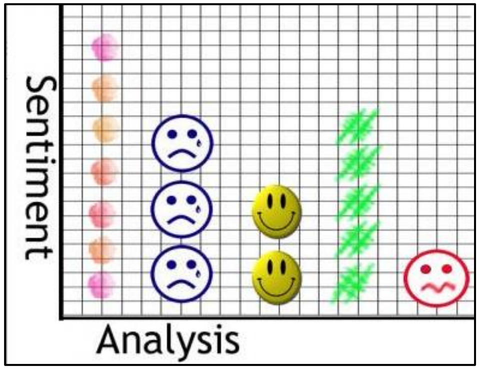
view: **All** | New Releases | Coming Soon

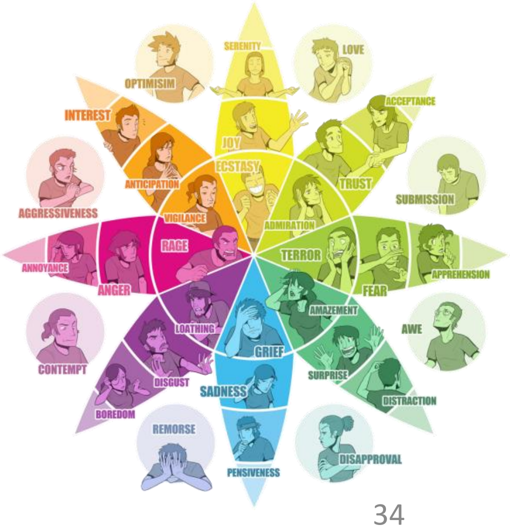
1. LOOK INSIDE!

Machine Learning in Action
by Peter Harrington (April 16, 2012)
Average Customer Review: ★★★½☆ (17)
In Stock

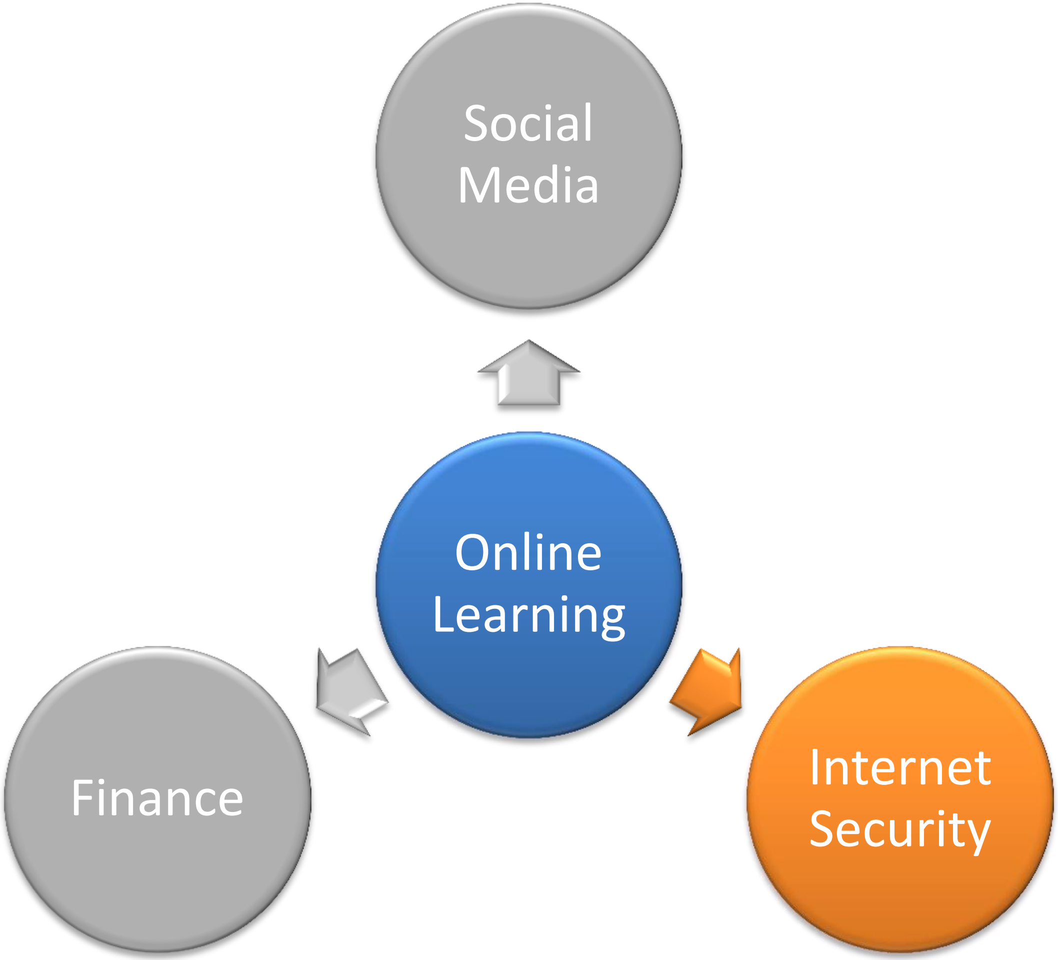
List Price: ~~$44.99~~
Price: $26.31
61 used & new from $20.00

☐ I own it ☐ Not interested ☒ ☆☆☆☆☆ Rate this item
Recommended because you purchased **Scaling up Machine Learning** and more (Fix this)





Where to Apply Online Learning?



Social Media

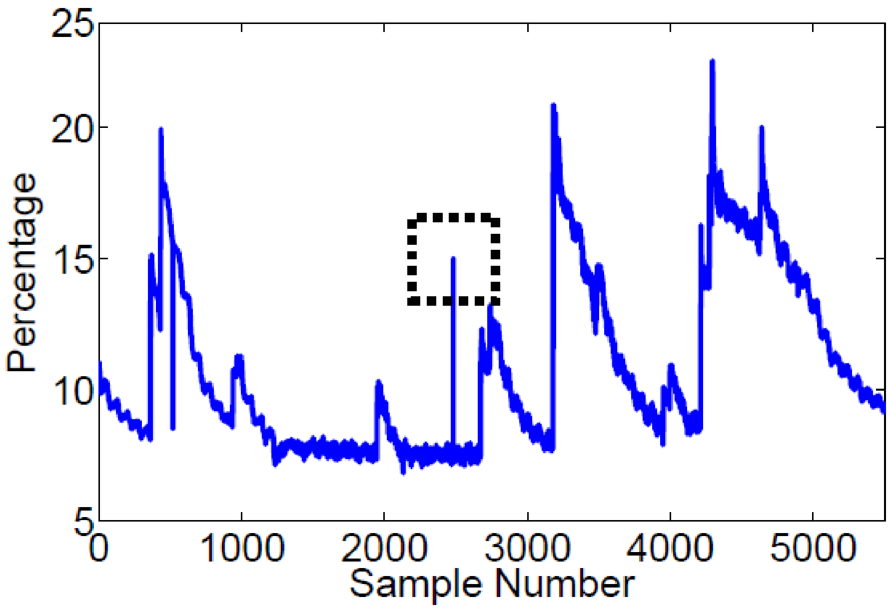
Online Learning

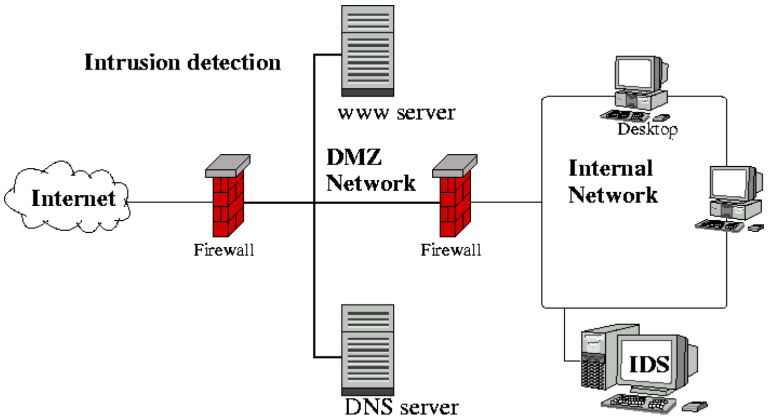
Finance

Internet Security

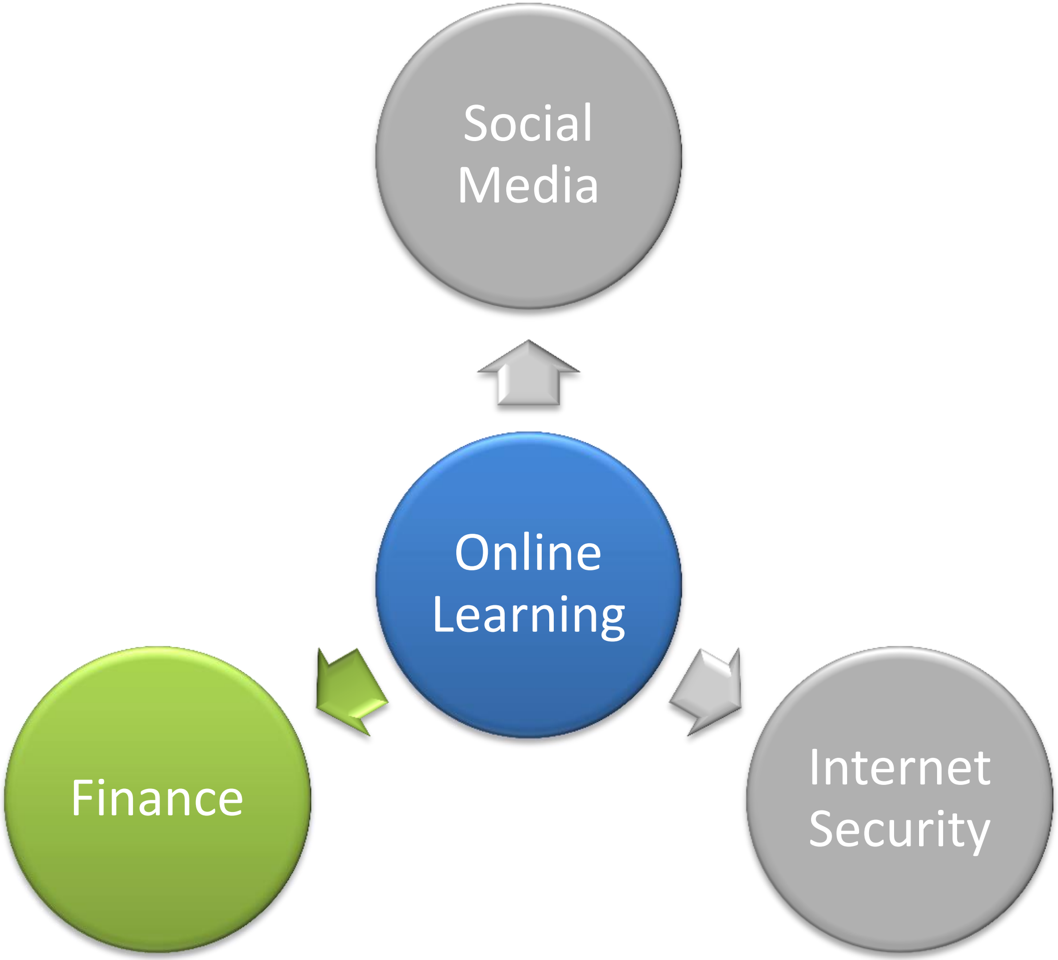
Online Learning for Internet Security

- Electronic business sectors
  - – **Spam email** filtering
  - – **Fraud** credit card transaction detection
  - – Network **intrusion detection** system, etc.



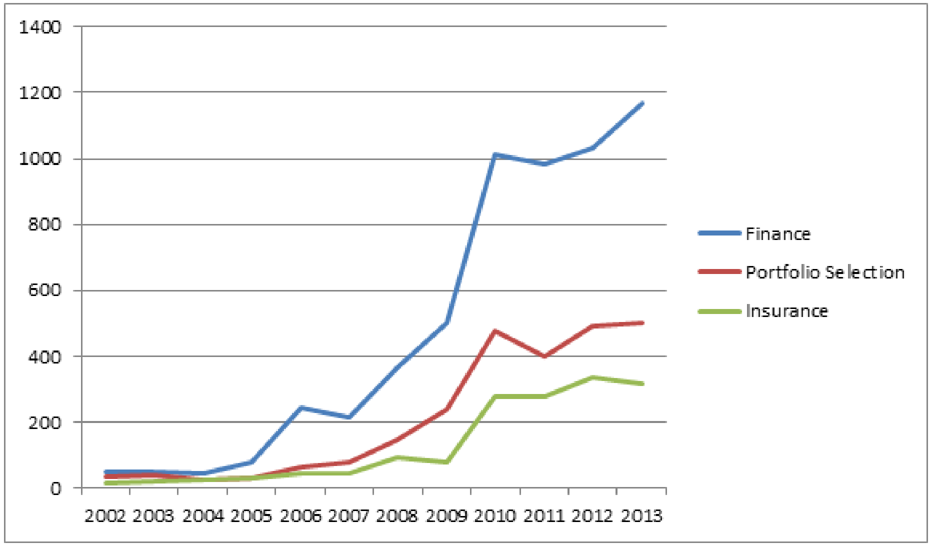


Where to Apply Online Learning?



Online Learning for Financial Decision

- Financial decision
  - Online portfolio selection
  - Sequential investment, etc.



Outline

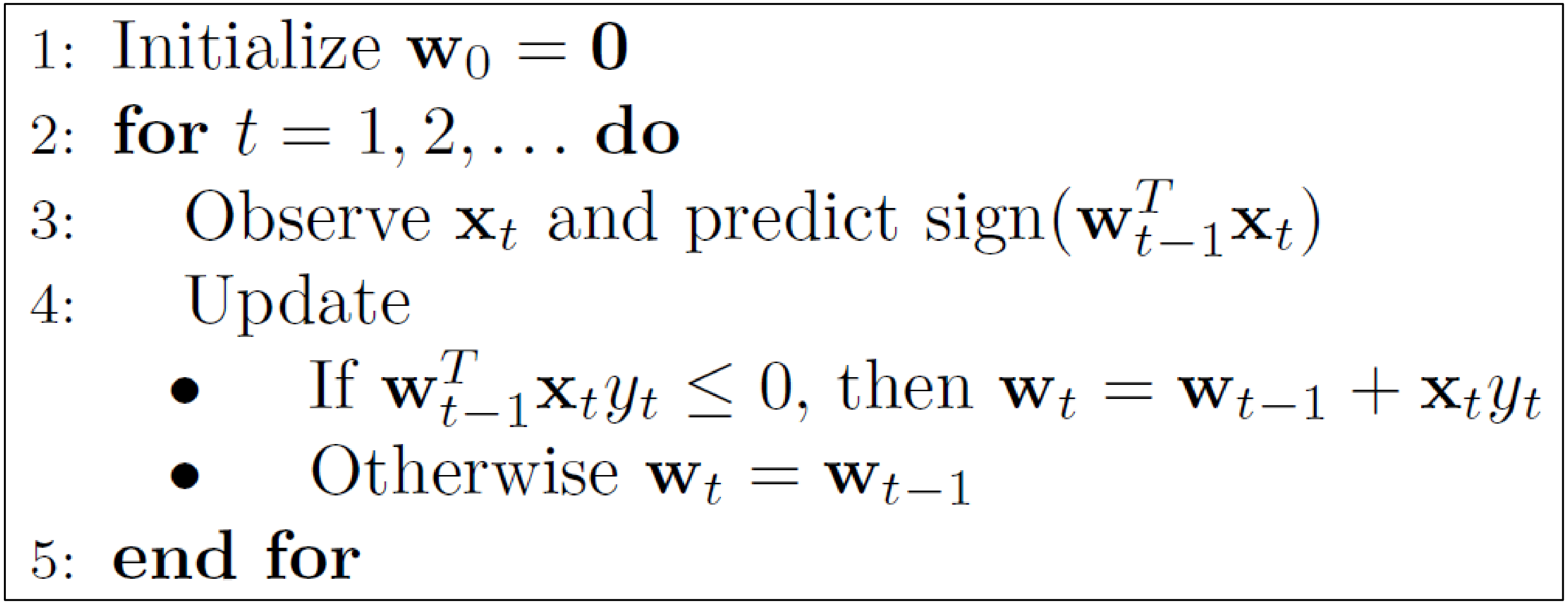
- Introduction (60 min.)
    - Big data and big data analytics (30 min.)
    - Online learning and its applications (30 min.)
- Online Learning Algorithms (60 min.)
    - Perceptron (10 min.)
    - Online non-sparse learning (10 min.)
    - Online sparse learning (20 min.)
    - Online unsupervised learning (20. min.)
- Discussions + Q & A (5 min.)

Outline

- Introduction (60 min.)
  - Big data and big data analytics (30 min.)
  - Online learning and its applications (30 min.)
- Online Learning Algorithms (60 min.)
  - Perceptron (10 min.)
  - Online non-sparse learning (10 min.)
  - Online sparse learning (20 min.)
  - Online unsupervised learning (20. min.)
- Discussions + Q & A (5 min.)

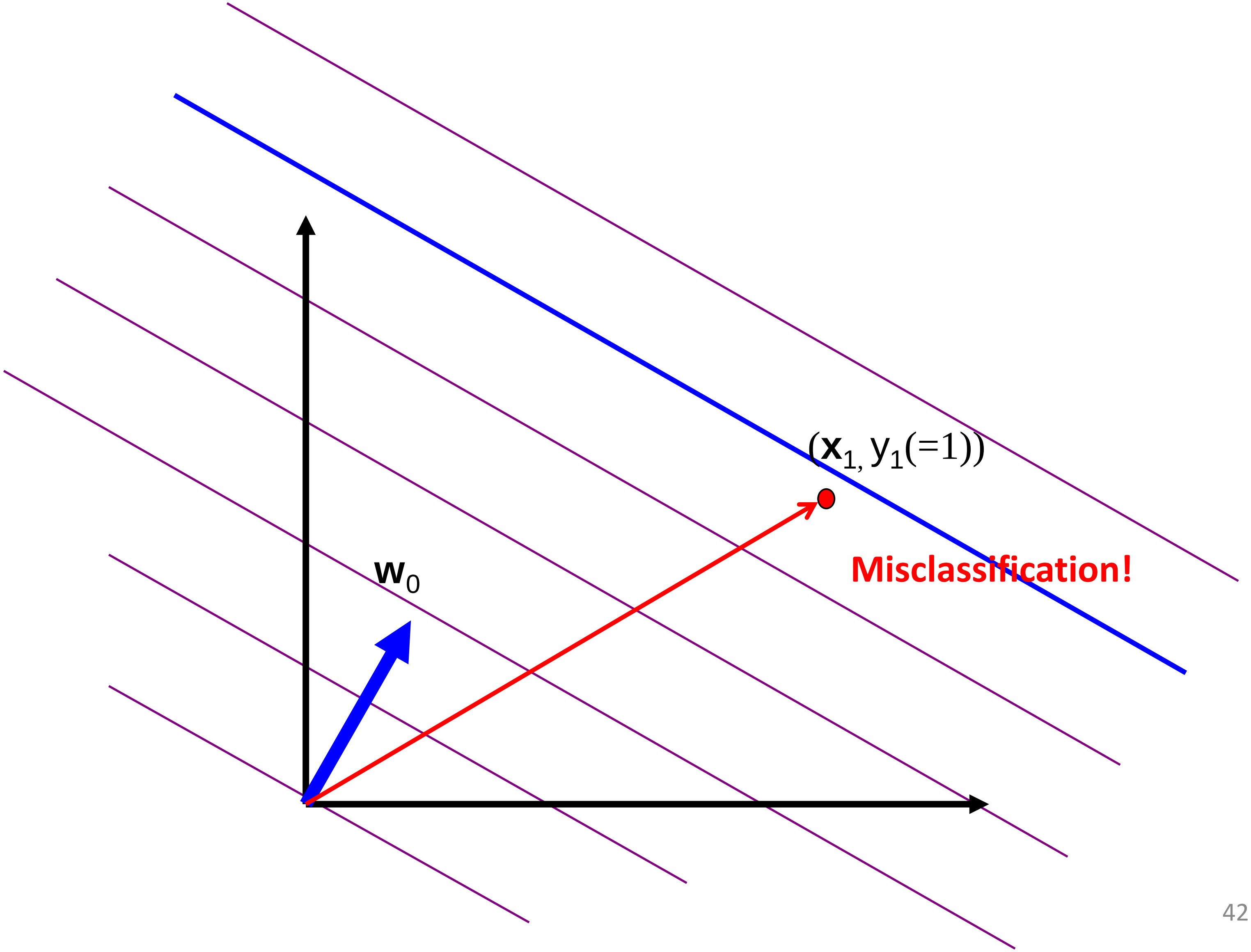
Perceptron Algorithm (F. Rosenblatt 1958)

- Goal: find a linear classifier with small error



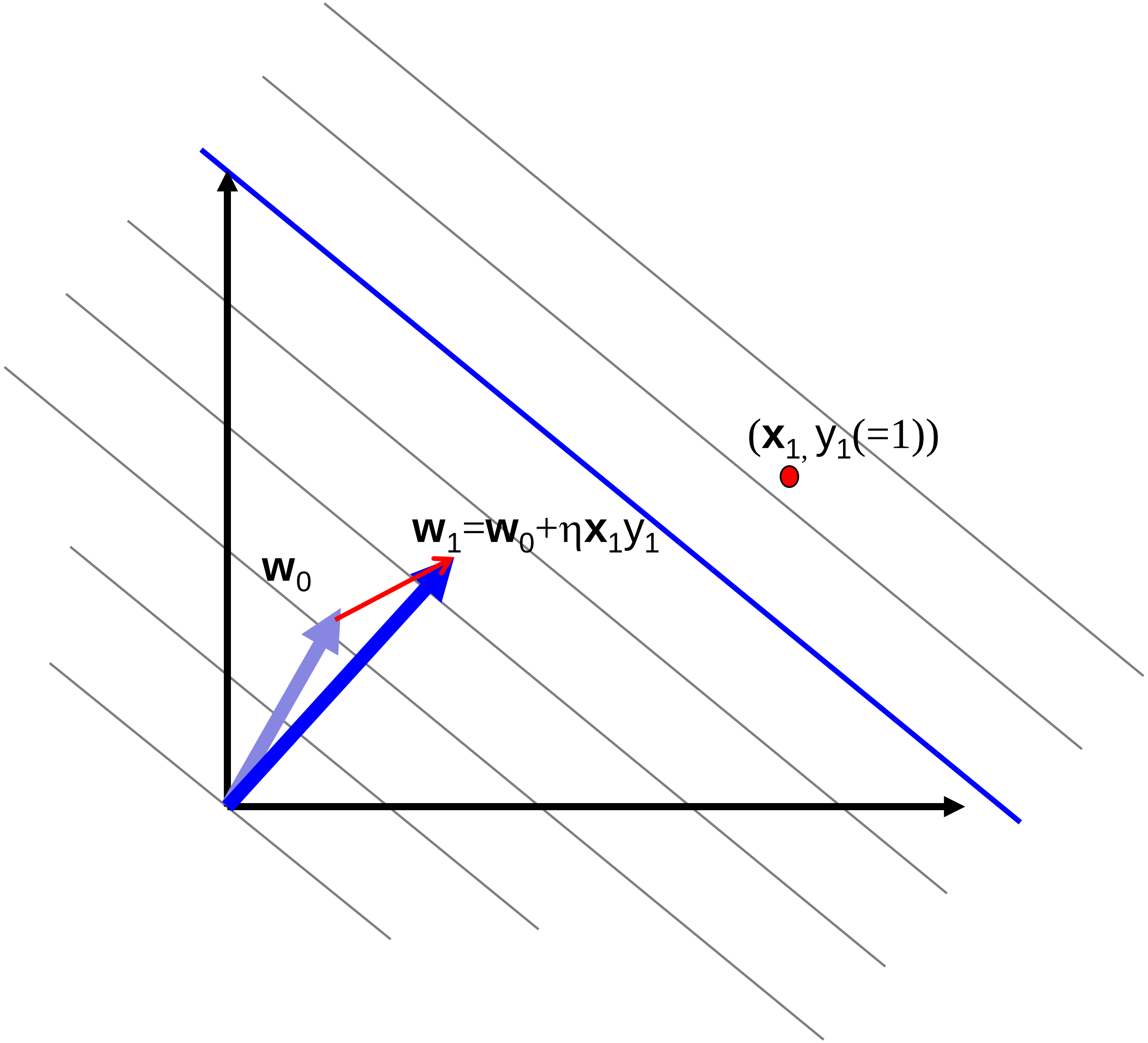
```
Initialize $\mathbf{w}_0 = \mathbf{0}$
for $t = 1, 2, \ldots$ do
    Observe $\mathbf{x}_t$ and predict $\mathrm{sign}(\mathbf{w}_{t-1}^T \mathbf{x}_t)$
    Update
        • If $\mathbf{w}_{t-1}^T \mathbf{x}_t y_t \leq 0$, then $\mathbf{w}_t = \mathbf{w}_{t-1} + \mathbf{x}_t y_t$
        • Otherwise $\mathbf{w}_t = \mathbf{w}_{t-1}$
end for
```

Geometric View

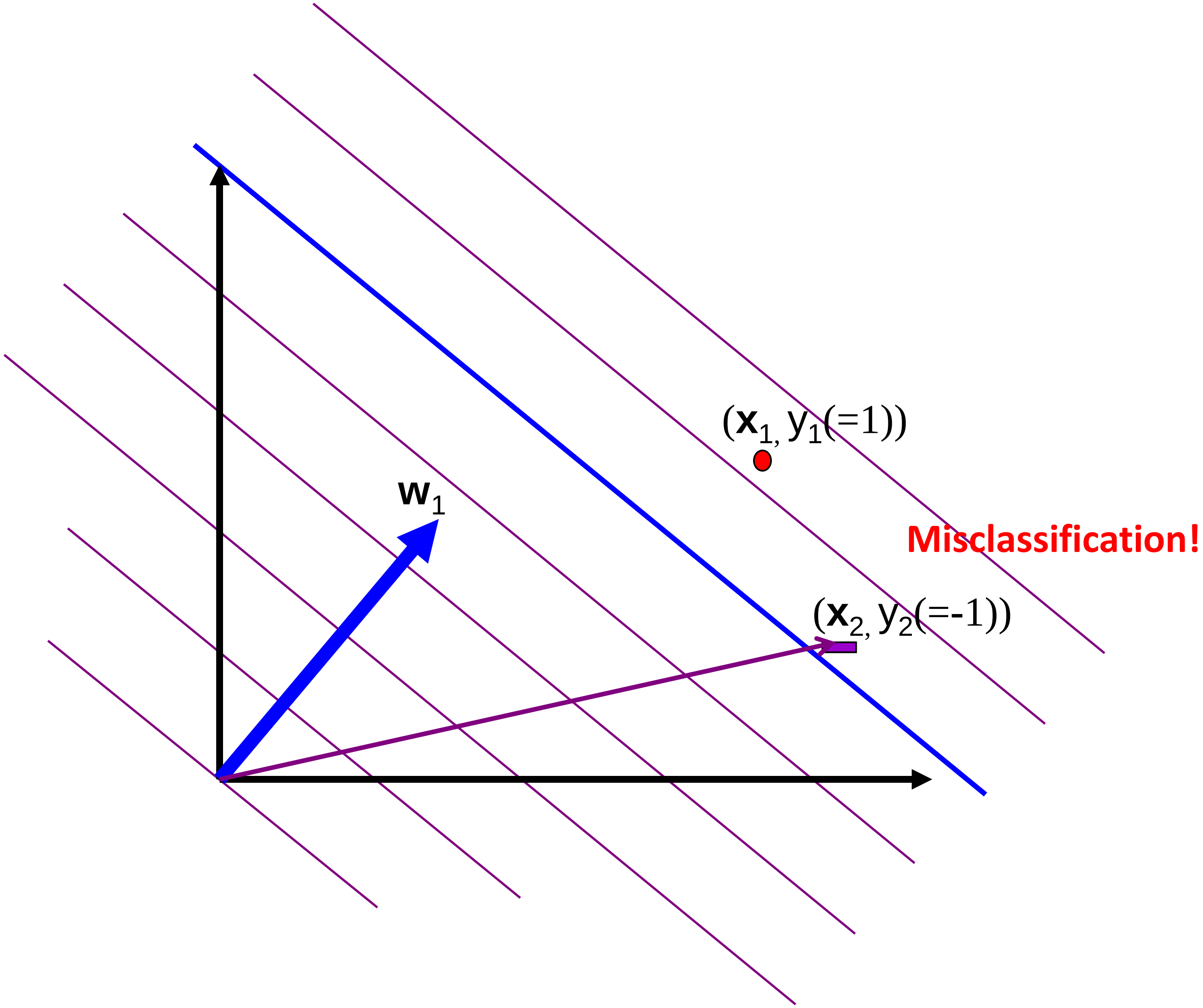


Misclassification!

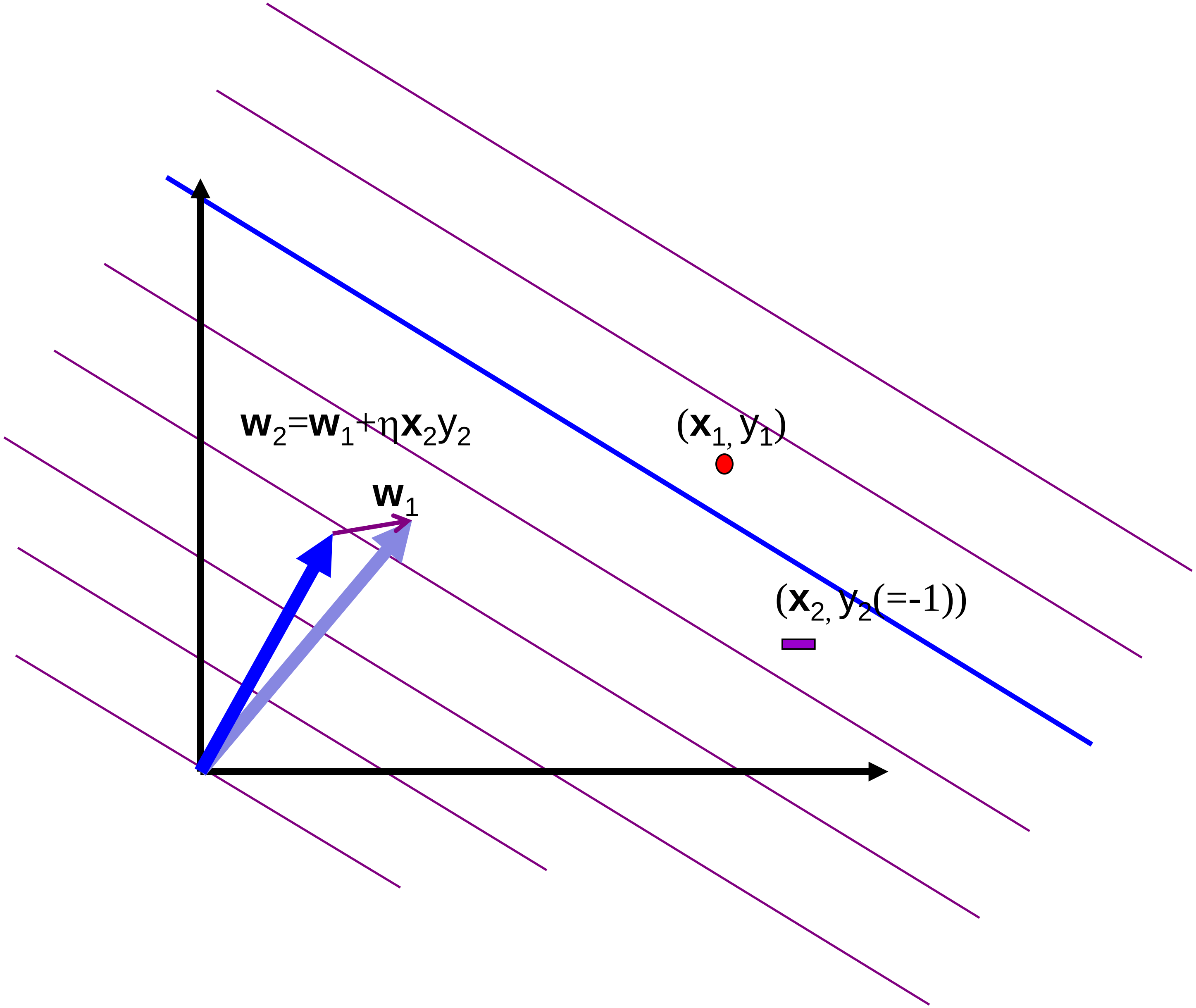
Geometric View



Geometric View



Geometric View



Perceptron Mistake Bound

- Consider w_* separate the data: $\mathbf{w}_*^T \mathbf{x}_i y_i > 0$
- Define margin

$$\gamma = \frac{\min_i \left| \mathbf{w}_*^T \mathbf{x}_i \right|}{\left\| \mathbf{w}_* \right\|_2 \sup_i \left\| \mathbf{x}_i \right\|_2}$$

- The number of mistakes perceptron makes is at most γ^{-2}

Proof of Perceptron Mistake Bound [Novikoff,1963]

Proof: Let $\mathbf{v}_k$ be the hypothesis before the k-th mistake. Assume that the k-th mistake occurs on the input example $(\mathbf{x}_i, y_i)$.

$$\gamma = \frac{\min_i |\mathbf{w}_*^T \mathbf{x}_i|}{\|\mathbf{w}_*\|_2 \sup_i \|\mathbf{x}_i\|_2}$$

First,

$$\begin{aligned} \|\mathbf{v}_{k+1}\|^2 &= \|\mathbf{v}_k + y_i \mathbf{x}_i\|^2 \\ &= \|\mathbf{v}_k\|^2 + 2y_i(\mathbf{v}_k^T \mathbf{x}_i) \\ &\quad + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{v}_k\|^2 + R^2 \\ &\leq kR^2 (R := \sup_i \|\mathbf{x}\|_2) \end{aligned}$$

Second,

$$\begin{aligned} \mathbf{v}_{k+1} &= \mathbf{v}_k + y_i \mathbf{x}_i \\ \mathbf{v}_{k+1}^T \mathbf{u} &= \mathbf{v}_k^T \mathbf{u} + y_i \mathbf{x}_i^T \mathbf{u} \\ &\geq \mathbf{v}_k^T \mathbf{u} + \gamma R \end{aligned}$$

$$\mathbf{v}_{k+1}^T \mathbf{u} \geq k\gamma R.$$

Hence,

$$\sqrt{k}R \geq \|\mathbf{v}_{k+1}\| \geq \mathbf{v}_{k+1}^T \mathbf{u} \geq k\gamma R$$

$$k \leq \gamma^{-2}$$

Outline

- Introduction (60 min.)
  - Big data and big data analytics (30 min.)
  - Online learning and its applications (30 min.)
- Online Learning Algorithms (60 min.)
  - Perceptron (10 min.)
  - Online non-sparse learning (10 min.)
  - Online sparse learning (20 min.)
  - Online unsupervised learning (20. min.)
- Discussions + Q & A (5 min.)

Online Non-Sparse Learning

- **First order** learning methods
  - **Online gradient descent** (Zinkevich, 2003)
  - **Passive aggressive learning** (Crammer et al., 2006)
  - Others (including but not limited)
    - ALMA: A New Approximate Maximal Margin Classification Algorithm (Gentile, 2001)
    - ROMMA: Relaxed Online Maximum Margin Algorithm (Li and Long, 2002)
    - MIRA: Margin Infused Relaxed Algorithm (Crammer and Singer, 2003)
    - DUOL: A Double Updating Approach for Online Learning (Zhao et al. 2009)

Online Gradient Descent (OGD)

- Online convex optimization (Zinkevich 2003)
  - Consider a convex objective function

    $$f : S \to \mathbb{R}$$

    where $S \subset \mathbb{R}^n$ is a bounded convex set
  - Update by Online Gradient Descent (OGD) or Stochastic Gradient Descent (SGD)

    $$\mathbf{w}_{t+1} \leftarrow \prod_S (\mathbf{w}_t - \eta \nabla f(\mathbf{w}_t))$$

    where η is a learning rate

Online Gradient Descent (OGD)

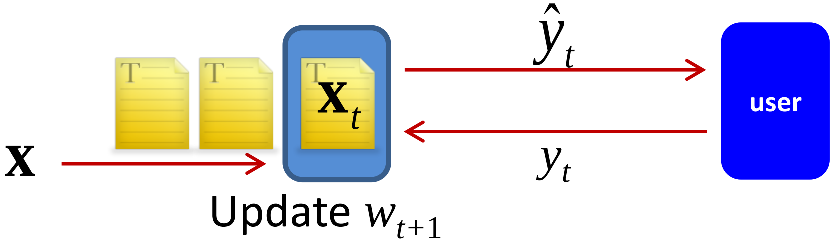
- For t=1, 2, …
  - An unlabeled sample $\mathbf{x}_t$ arrives
  - Make a prediction based on existing weights

$$\hat{y}_t = \mathrm{sgn}(\mathbf{w}_t^T \mathbf{x}_t)$$

  - Observe the true class label $y_t \in \{-1,+1\}$
  - Update the weights by

$$\mathbf{w}_{t+1} \leftarrow \prod_S (\mathbf{w}_t - \eta \nabla f(\mathbf{w}_t))$$

where η is a learning rate



Passive Aggressive Online Learning

- Closed-form solutions can be derived:

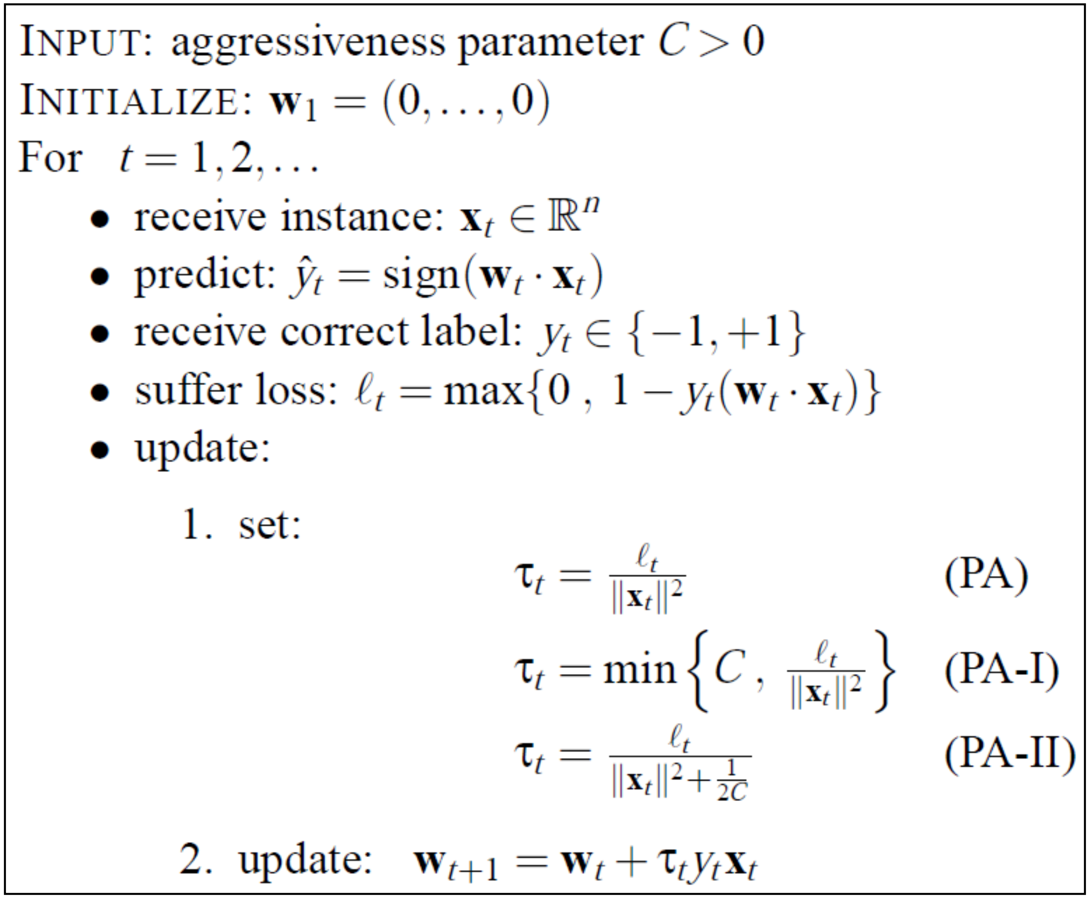
$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$$

$$\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2} \qquad \text{(PA)}$$

$$\tau_t = \min\left\{C\,,\ \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\} \qquad \text{(PA-I)}$$

$$\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}} \qquad \text{(PA-II)}$$

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell(\mathbf{w};(\mathbf{x}_t, y_t)) = 0$$



```
INPUT: aggressiveness parameter $C > 0$
INITIALIZE: $\mathbf{w}_1 = (0,\ldots,0)$
For $t = 1,2,\ldots$
    • receive instance: $\mathbf{x}_t \in \mathbb{R}^n$
    • predict: $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$
    • receive correct label: $y_t \in \{-1,+1\}$
    • suffer loss: $\ell_t = \max\{0\ ,\ 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)\}$
    • update:
        1. set:
            $\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2}$    (PA)
            $\tau_t = \min\left\{C\,,\ \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\}$    (PA-I)
            $\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}}$    (PA-II)
        2. update: $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$
```

Online Non-Sparse Learning

- **First order** methods
  - Learn a **linear** weight vector (first order) of model
- Pros and Cons
  - 😆 Simple and easy to implement
  - 😆 Efficient and scalable for high-dimensional data
  - 😭 Relatively slow convergence rate

Online Non-Sparse Learning

- **Second order** online learning methods
  - Update the weight vector **w** by maintaining and exploring both first-order and second-order information
- Some representative methods, but not limited
  - SOP: Second Order Perceptron (Cesa-Bianchi et al., 2005)
  - CW: Confidence Weighted learning (Dredze et al., 2008)
  - AROW: Adaptive Regularization of Weights (Crammer et al., 2009)
  - IELLIP: Online Learning by Ellipsoid Method (Yang et al., 2009)
  - NHERD: Gaussian Herding (Crammer & Lee 2010)
  - NAROW: New variant of AROW algorithm (Orabona & Crammer 2010)
  - SCW: Soft Confidence Weighted (SCW) (Hoi et al., 2012)

Online Non-Sparse Learning

- **Second-Order** online learning methods
  - Learn the weight vector **w** by maintaining and exploring both first-order and second-order information
- Pros and Cons
  - 😆 Faster convergence rate
  - 😫 Expensive for high-dimensional data
  - 😫 Relatively sensitive to noise

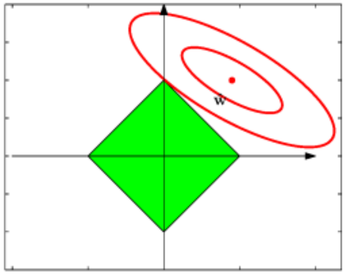
Outline

- Introduction (60 min.)
  - Big data and big data analytics (30 min.)
  - Online learning and its applications (30 min.)
- Online Learning Algorithms (60 min.)
  - Perceptron (10 min.)
  - Online non-sparse learning (10 min.)
  - Online sparse learning (20 min.)
  - Online unsupervised learning (20. min.)
- Discussions + Q & A (5 min.)

Online Sparse Learning

- Motivation
  - Space constraint: RAM overflow
  - Test-time constraint
  - How to induce Sparsity in the weights of online learning algorithms?
- Some representative work
  - **Truncated gradient** (Langford et al., 2009)
  - **FOBOS**: Forward Looking Subgradients (Duchi and Singer 2009)
  - **Dual averaging** (Xiao, 2009)
  - etc.

Truncated Gradient (Langford et al., 2009)



- Objective function

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{n} L(w, z_i) + g\|w\|_1$$

- Stochastic gradient descent

$$f(w_i) = w_i - \eta \nabla_1 L(w_i, z_i)$$

- Simple coefficient rounding

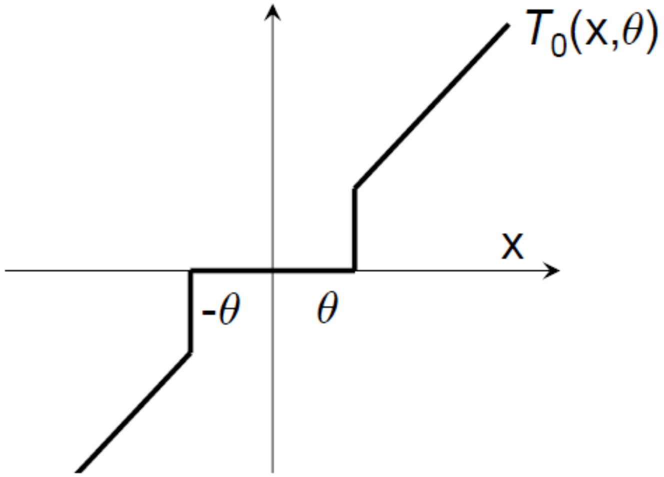
$$f(w_i) = T_0(w_i - \eta \nabla_1 L(w_i, z_i), \theta)$$

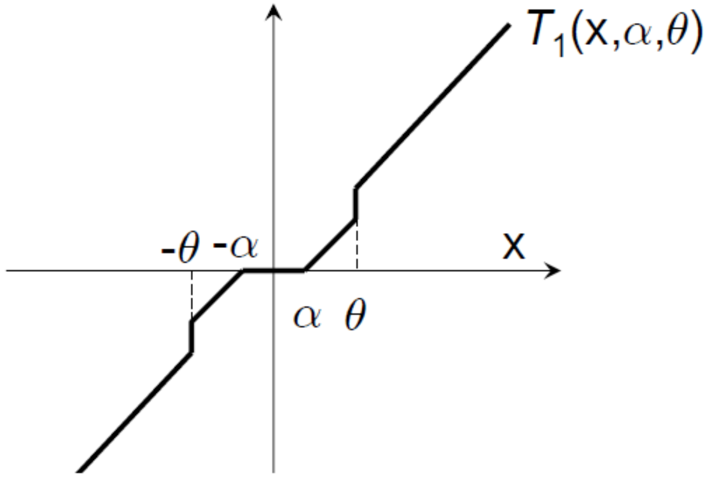
Truncated gradient: impose sparsity by modifying the stochastic gradient descent

Truncated Gradient (Langford et al., 2009)

Simple Coefficient Rounding vs. Less aggressive truncation

$$T_0(v_j, \theta) = \begin{cases} 0 & \text{if } |v_j| \leq \theta \\ v_j & \text{otherwise} \end{cases} \quad T_1(v_j, \alpha, \theta) = \begin{cases} \max(0, v_j - \alpha) & \text{if } v_j \in [0, \theta] \\ \min(0, v_j + \alpha) & \text{if } v_j \in [-\theta, 0] \\ v_j & \text{otherwise} \end{cases}$$



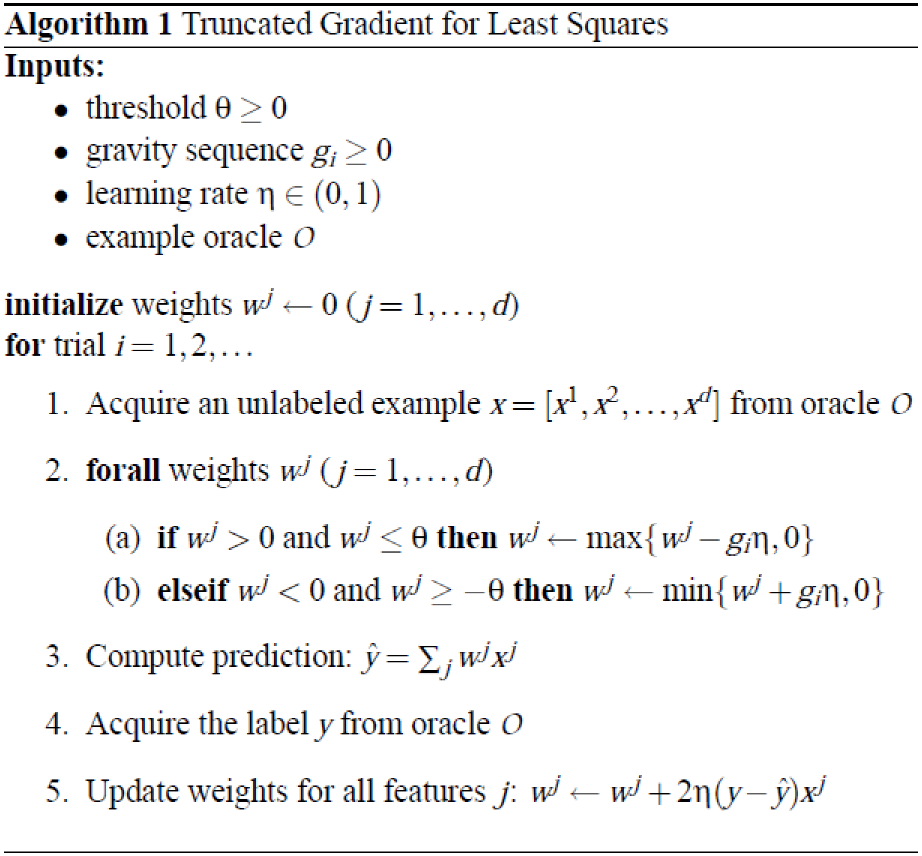


Truncated Gradient (Langford et al., 2009)

$$f(w_i) = T_1(w_i - \eta \nabla_1 L(w_i, z_i), \eta g_i, \theta)$$

- The amount of shrinkage is measured by a *gravity* parameter $g_i > 0$
- The truncation can be performed every K online steps
- When $g_i = 0$

  the update rule is identical to the standard SGD
- Loss functions: $L(w, z) = \phi(w^T x, y)$
  - Logistic $\phi(p, y) = \ln(1 + \exp(-py))$
  - SVM (hinge) $\phi(p, y) = \max(0, 1 - py)$
  - Least square $\phi(p, y) = (p - y)^2$



```
Algorithm 1 Truncated Gradient for Least Squares
Inputs:
  • threshold $\theta \geq 0$
  • gravity sequence $g_i \geq 0$
  • learning rate $\eta \in (0, 1)$
  • example oracle $O$

initialize weights $w^j \leftarrow 0$ $(j = 1, \ldots, d)$
for trial $i = 1, 2, \ldots$
  1. Acquire an unlabeled example $x = [x^1, x^2, \ldots, x^d]$ from oracle $O$
  2. forall weights $w^j$ $(j = 1, \ldots, d)$
       (a) if $w^j > 0$ and $w^j \leq \theta$ then $w^j \leftarrow \max\{w^j - g_i\eta, 0\}$
       (b) elseif $w^j < 0$ and $w^j \geq -\theta$ then $w^j \leftarrow \min\{w^j + g_i\eta, 0\}$
  3. Compute prediction: $\hat{y} = \sum_j w^j x^j$
  4. Acquire the label $y$ from oracle $O$
  5. Update weights for all features $j$: $w^j \leftarrow w^j + 2\eta(y - \hat{y})x^j$
```

Truncated Gradient (Langford et al., 2009)

- Regret bound

$$\frac{1-0.5A\eta}{T}\sum_{i=1}^{T}\left[L(w_i,z_i)+\frac{g_i}{1-0.5A\eta}\|w_{i+1}\cdot I(w_{i+1}\le\theta)\|_1\right]$$

$$\le\frac{\eta}{2}B+\frac{\|\bar{w}\|^2}{2\eta T}+\frac{1}{T}\sum_{i=1}^{T}[L(\bar{w},z_i)+g_i\|\bar{w}\cdot I(w_{i+1}\le\theta)\|_1],$$

- Let $\eta = 1/\sqrt{T}$

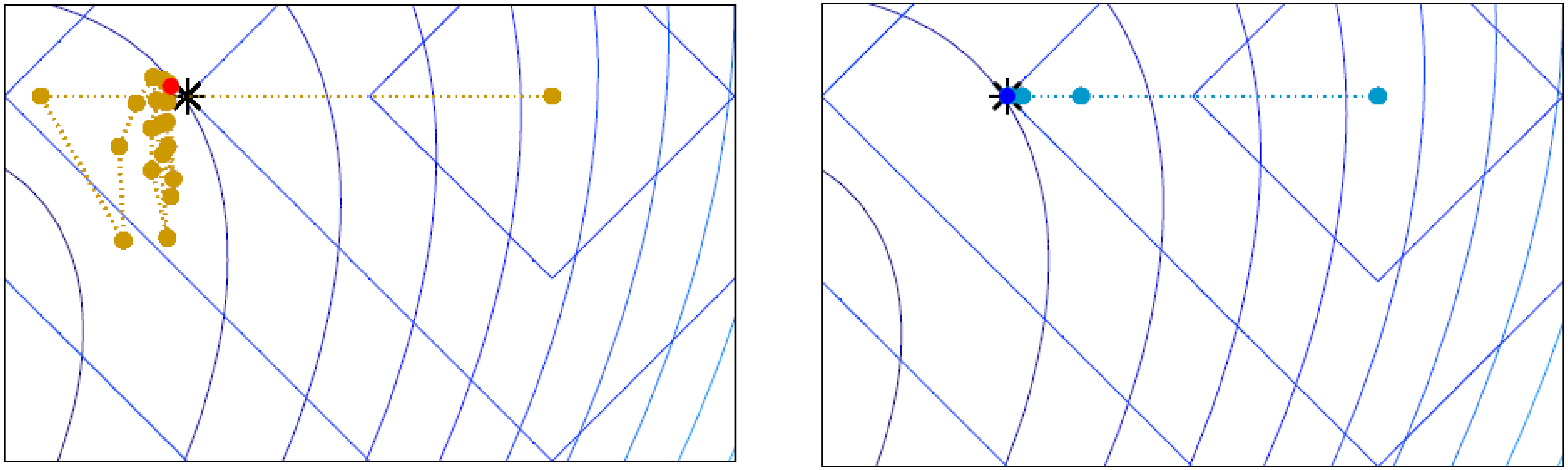
$$\sum_{i=1}^{T}(L(w_i,z_i)+g\|w_i\|_1)-\sum_{i=1}^{T}(L(\bar{w},z_i)+g\|\bar{w}\|_1)$$

$$\le\frac{\sqrt{T}}{2}(B+\|\bar{w}\|^2)\left(1+\frac{A}{2\sqrt{T}}\right)+\frac{A}{2\sqrt{T}}\left(\sum_{i=1}^{T}L(\bar{w},z_i)+g\sum_{i=1}^{T}(\|\bar{w}\|_1-\|w_{i+1}\|_1)\right)+o(\sqrt{T})$$

FOBOS (Duchi & Singer, 2009)

- FOrward-Backward Splitting

Minimize $\frac{1}{2}\boldsymbol{w}^\top A\boldsymbol{w} + \boldsymbol{c}^\top\boldsymbol{w} + \lambda\|\boldsymbol{w}\|_1$. True solution: $\boldsymbol{w}^* = [-1\ 0]^\top$.



Subgradient

Fobos

FOBOS (Duchi & Singer, 2009)

- Objective function

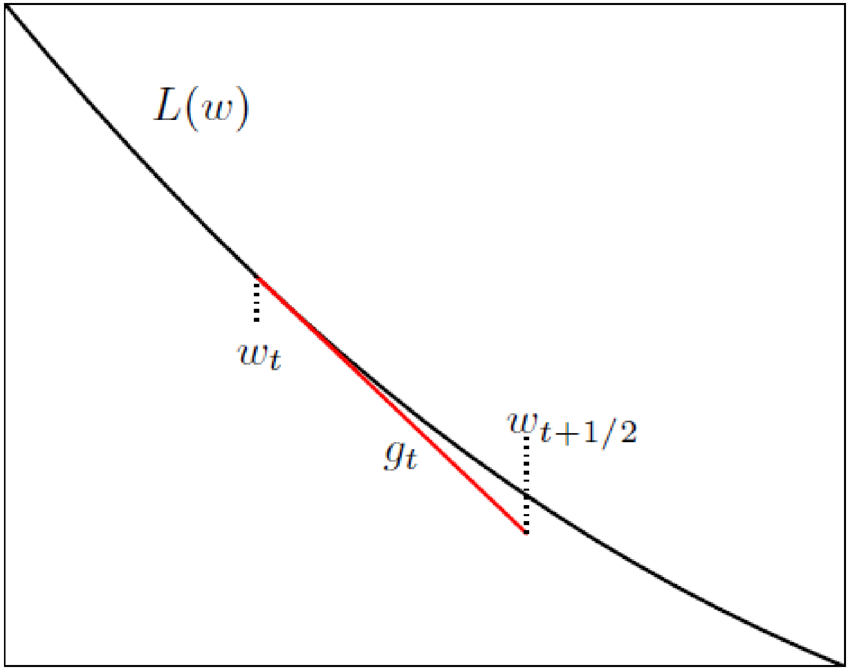
$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) + R(\boldsymbol{w})$$

- Repeat
  I. Unconstrained (stochastic sub) gradient of loss
  II. Incorporate regularization
- Similar to
  - Forward-backward splitting (Lions and Mercier 79)
  - Composite gradient methods (Wright et al. 09, Nesterov 07)

FOBOS: Step I

- Objective function $\min_{\boldsymbol{w}} L(\boldsymbol{w}) + R(\boldsymbol{w})$
- Unconstrained (stochastic sub) gradient of loss

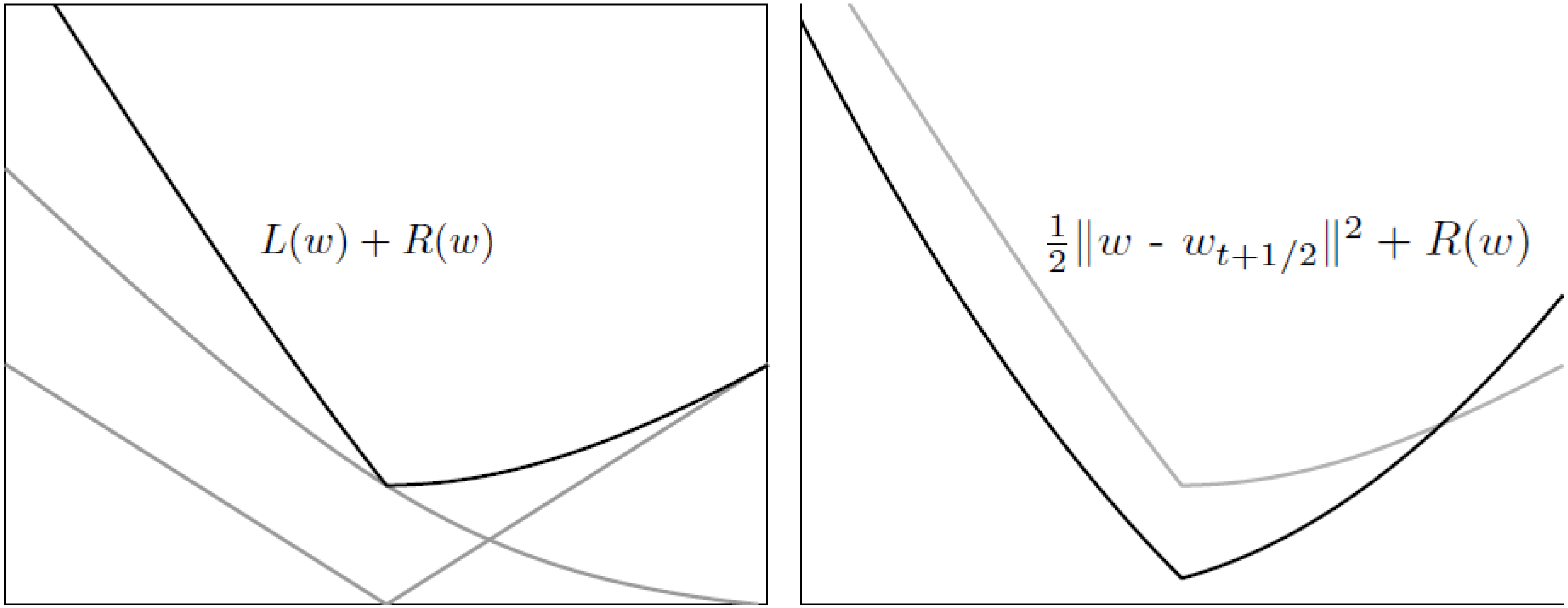
$$\boldsymbol{w}_{t+\frac{1}{2}} = \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t \quad \text{where} \quad \mathbb{E}\boldsymbol{g}_t \in \partial L(\boldsymbol{w}_t)$$



FOBOS: Step II

- Objective function $\min_{\boldsymbol{w}} L(\boldsymbol{w}) + R(\boldsymbol{w})$
- Incorporate regularization

$$\boldsymbol{w}_{t+1} = \operatorname*{argmin}_{\boldsymbol{w}} \left\{ \frac{1}{2} \left\| \boldsymbol{w} - \boldsymbol{w}_{t+\frac{1}{2}} \right\|^2 + \eta_t R(\boldsymbol{w}) \right\}$$



Forward Looking Property

- The optimum $\boldsymbol{w}_{t+1}$ satisfies

$$0 \in \boldsymbol{w}_{t+1} - \boldsymbol{w}_t + \eta_t \partial L(\boldsymbol{w}_t) + \eta_t \partial R(\boldsymbol{w}_{t+1})$$

- Let $\boldsymbol{g}_t^L \in \partial L(\boldsymbol{w}_t)$ and $\boldsymbol{g}_{t+1}^R \in \partial R(\boldsymbol{w}_{t+1})$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t^L - \eta_t \boldsymbol{g}_{t+1}^R$$

current loss forward regularization

- *Current* subgradient of loss, *forward* subgradient of regularization

Batch Convergence and Online Regret

- Set $\eta_t \propto \frac{1}{\sqrt{T}}$ or $\frac{1}{\sqrt{t}}$ to obtain batch convergence

$$L(\boldsymbol{w}_t) + R(\boldsymbol{w}_t) - (L(\boldsymbol{w}^*) + R(\boldsymbol{w}^*)) = O\left(\frac{1}{\sqrt{T}}\right)$$

- Online (average) regret bounds

$$\text{Regret}(T) \triangleq \frac{1}{T}\left[\sum_{t=1}^{T} L_t(\boldsymbol{w}_t) + R(\boldsymbol{w}_t) - \sum_{t=1}^{T} L_t(\boldsymbol{w}^*) + R(\boldsymbol{w}^*)\right]$$

$$\eta_t \propto \frac{1}{\sqrt{t}} \quad \Rightarrow \quad \text{Regret}(T) = O\left(\frac{1}{\sqrt{T}}\right)$$

$$\eta_t \propto \frac{1}{t} \quad \Rightarrow \quad \text{Regret}(T) = O\left(\frac{\log T}{T}\right) \quad \text{(strong convexity)}$$

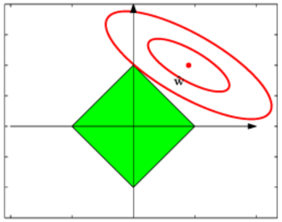
High Dimensional Efficiency

- Input space is sparse but huge
- Need to perform lazy updates to $\boldsymbol{w}$
- **Proposition:** The following are equivalent

$$\boldsymbol{w}_t = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w} - \boldsymbol{w}_{t-1}\|^2 + \eta_t \lambda \|\boldsymbol{w}\|_q \quad \text{for } t = 1 \text{ to } T$$

$$\boldsymbol{w}_T = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w} - \boldsymbol{w}_0\|^2 + \left(\sum_{t=1}^{T-1} \eta_t \lambda \right) \|\boldsymbol{w}\|_q$$

Dual Averaging (Xiao, 2010)



- Objective function

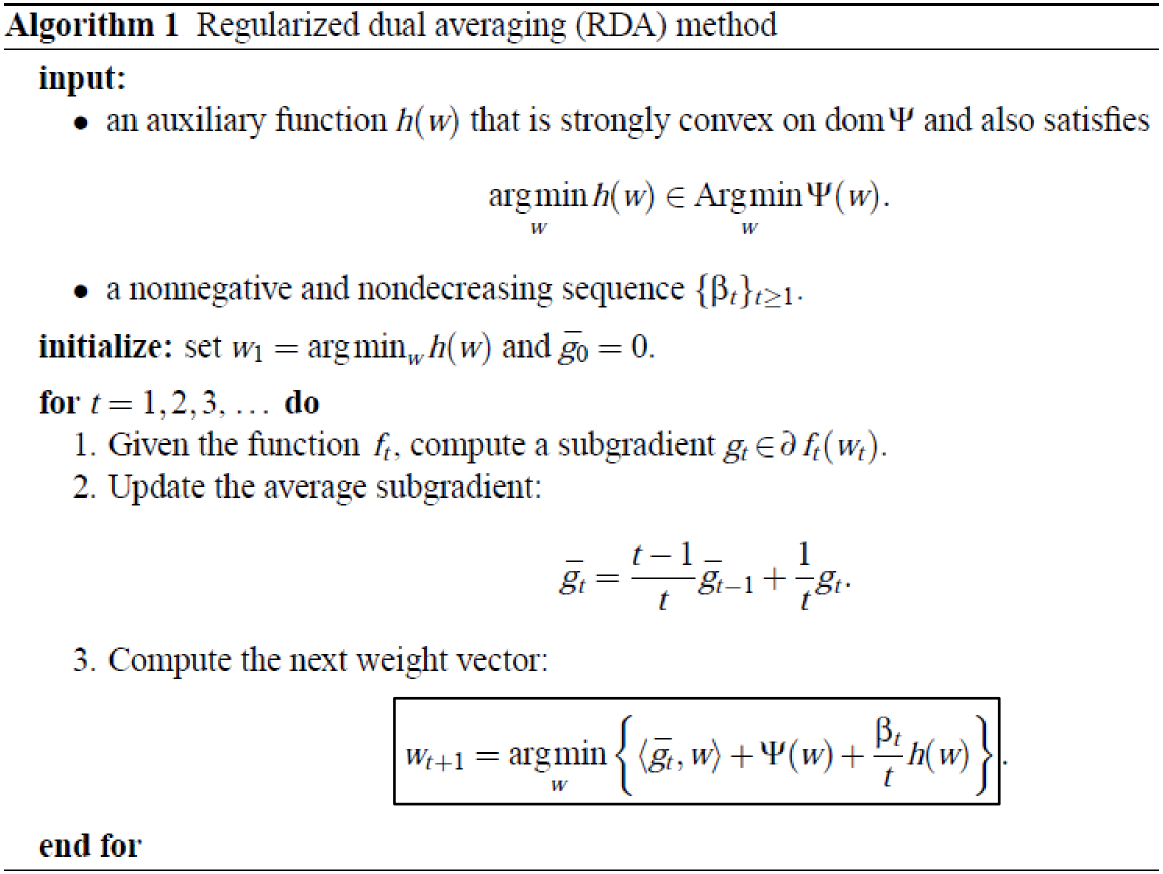
$$\underset{w}{\text{minimize}} \quad \left\{ \phi(w) \triangleq \mathbf{E}_z f(w,z) + \Psi(w) \right\} \quad \Psi(w) = \lambda \| w \|_1 \text{ with } \lambda > 0$$

- Problem: truncated gradient doesn’t produce truly sparse weight due to small learning rate
- Fix: dual averaging which keeps two state representations:
  - – parameter w_t and average gradient vector

$$\bar{g}_t = \frac{1}{t} \sum_{i=1}^{t} f_i(w_i)$$

Dual Averaging (Xiao, 2010)

- w_{t+1} has entry-wise closed-form solution
- Advantage: sparse on the weight w_t
- Disadvantage: keep a non-sparse subgradient $\bar{g}_t$



Algorithm 1 Regularized dual averaging (RDA) method

```
input:
    • an auxiliary function $h(w)$ that is strongly convex on $\mathrm{dom}\,\Psi$ and also satisfies
          $\arg\min_w h(w) \in \mathrm{Arg}\min_w \Psi(w).$
    • a nonnegative and nondecreasing sequence $\{\beta_t\}_{t\geq 1}$.
initialize: set $w_1 = \arg\min_w h(w)$ and $\bar{g}_0 = 0$.
for $t = 1, 2, 3, \ldots$ do
    1. Given the function $f_t$, compute a subgradient $g_t \in \partial f_t(w_t)$.
    2. Update the average subgradient:
          $\bar{g}_t = \frac{t-1}{t}\bar{g}_{t-1} + \frac{1}{t} g_t.$
    3. Compute the next weight vector:
          $\boxed{w_{t+1} = \arg\min_w \left\{ \langle \bar{g}_t, w \rangle + \Psi(w) + \frac{\beta_t}{t} h(w) \right\}}.$
end for
```

$$w_{t+1}^{(i)} = \begin{cases} 0 & \text{if } \left|\bar{g}_t^{(i)}\right| \leq \lambda, \\ -\frac{\sqrt{t}}{\gamma}\left(\bar{g}_t^{(i)} - \lambda\,\mathrm{sgn}\left(\bar{g}_t^{(i)}\right)\right) & \text{otherwise,} \end{cases}$$

Convergence and Regret

- Average regret

$$\bar{R}_T(w) \triangleq \frac{1}{T}\sum_{t=1}^{T}(f_t(w_t) + \Psi(w_t)) - S_T(w)$$

$$S_T(w) \triangleq \frac{1}{T}\sum_{t=1}^{T}(f_t(w) + \Psi(w))$$

- Theoretical bound: similar to gradient descent

$$\bar{R}_T \sim \mathcal{O}(1/\sqrt{T})$$

$$\bar{R}_T \sim \mathcal{O}(\log(T)/T), \quad \text{if } h(\cdot) \text{ is strongly convex}$$

Comparison

- FOBOS

$$w_{t+1} = \arg\min_{w} \left\{ \langle g_t, w \rangle + \Psi(w) + \frac{1}{2\alpha_t} \|w - w_t\|_2^2 \right\}$$

  - Subgradient g_t
  - Local Bregman divergence
  - Coefficient $1/\alpha_t = \Theta(\sqrt{t})$
- Equivalent to TG method when $\Psi(w) = \|w\|_1$

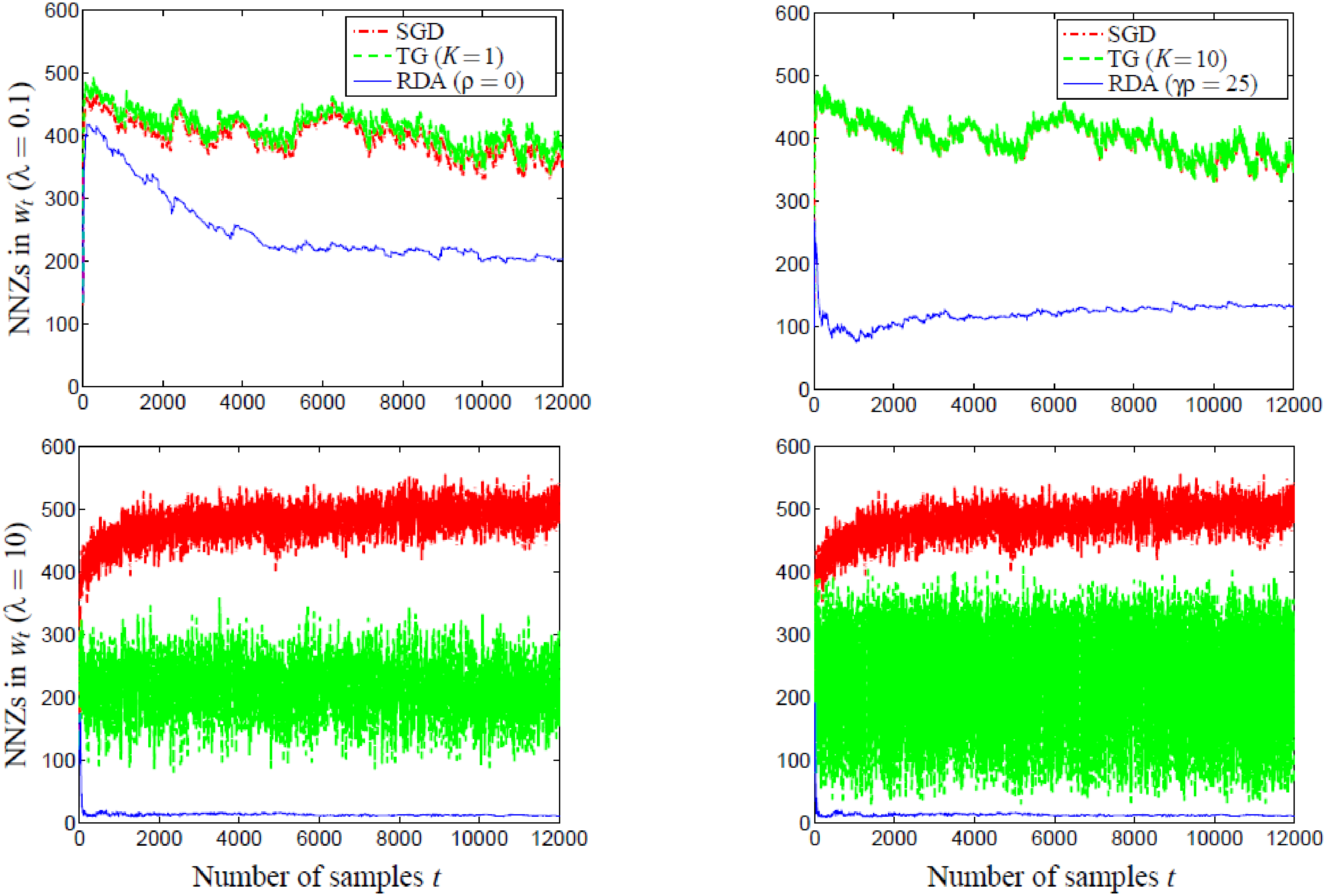
- Dual Averaging

$$w_{t+1} = \arg\min_{w} \left\{ \langle \bar{g}_t, w \rangle + \Psi(w) + \frac{\beta_t}{t} h(w) \right\}$$

  - Average subgradient $\bar{g}_t$
  - Global proximal function
  - Coefficient $\beta_t / t = \Theta\left(1/\sqrt{t}\right)$

Comparison

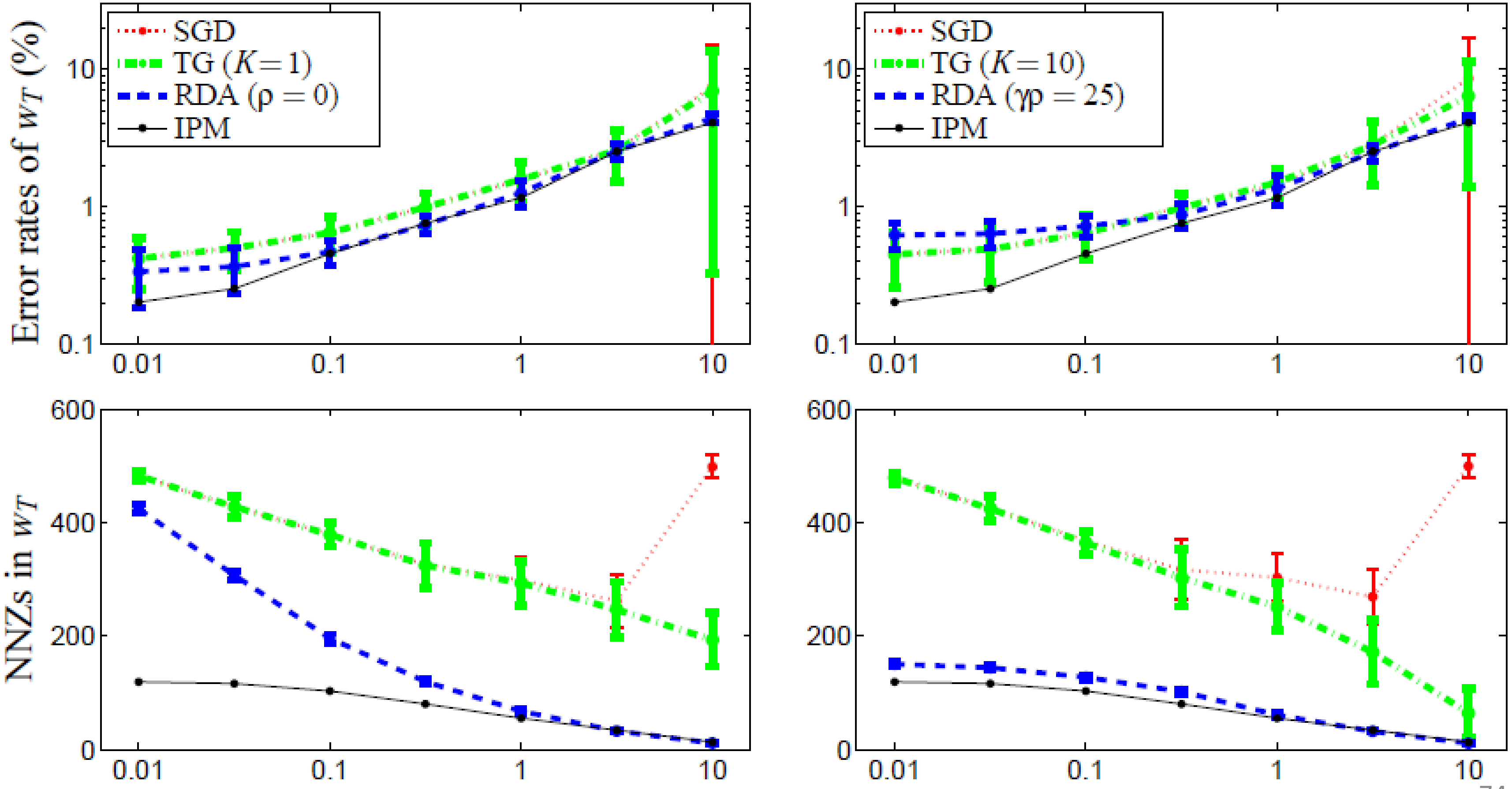
Left:$K = 1$ for TG, $\rho = 0$ for RDA   Right:K=10 for TG, $\gamma\rho$=25 for RDA



Comparison

Left:$K = 1$ for TG, $\rho = 0$ for RDA

Right:K=10 for TG, $\gamma\rho$=25 for RDA



Variants of Online Sparse Learning Models

- Online feature selection (OFS)
  - A variant of sparse online learning
  - The key difference is that OFS focuses on selecting a fixed subset of features in online learning process
  - Could be used as an alternative tool for batch feature selection when dealing with big data
- Other existing work
  - Online learning for Group Lasso (Yang et al., 2010) and online learning for multi-task feature selection (Yang et al. 2013) to select features in group manner or features among similar tasks

Online Sparse Learning

- Objective
  - Induce sparsity in the weights of online learning algorithms
- Pros and Cons

😆 Simple and easy to implement

😆 Efficient and scalable for high-dimensional data

😭 Relatively slow convergence rate

😭 No perfect way to attain sparsity solution yet

Outline

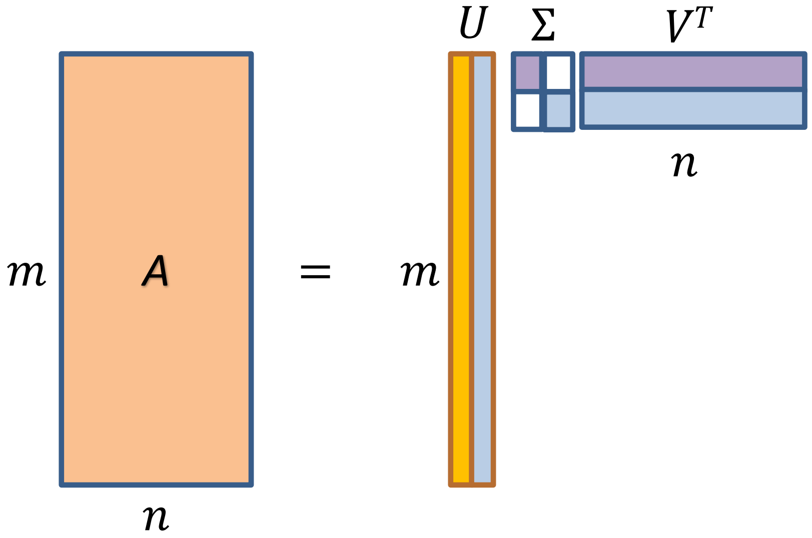
- Introduction (60 min.)
  - Big data and big data analytics (30 min.)
  - Online learning and its applications (30 min.)
- Online Learning Algorithms (60 min.)
  - Perceptron (10 min.)
  - Online non-sparse learning (10 min.)
  - Online sparse learning (20 min.)
  - Online unsupervised learning (20. min.)
- Discussions + Q & A (5 min.)

Online Unsupervised Learning

- Assumption: data generated from some underlying parametric probabilistic density function
- Goal: estimate the parameters of the density to give a suitable compact representation
- Typical work
  - **Online singular value decomposition** (SVD) (Brand, 2003)
- Others (including but not limited)
  - Online principal component analysis (PCA) (Warmuth and Kuzmin, 2006)
  - Online dictionary learning for sparse coding (Mairal et al. 2009)
  - Online learning for latent Dirichlet allocation (LDA) (Hoffman et al., 2010)
  - Online variational inference for the hierarchical Dirichlet process (HDP) (Wang et al. 2011)
  - ...

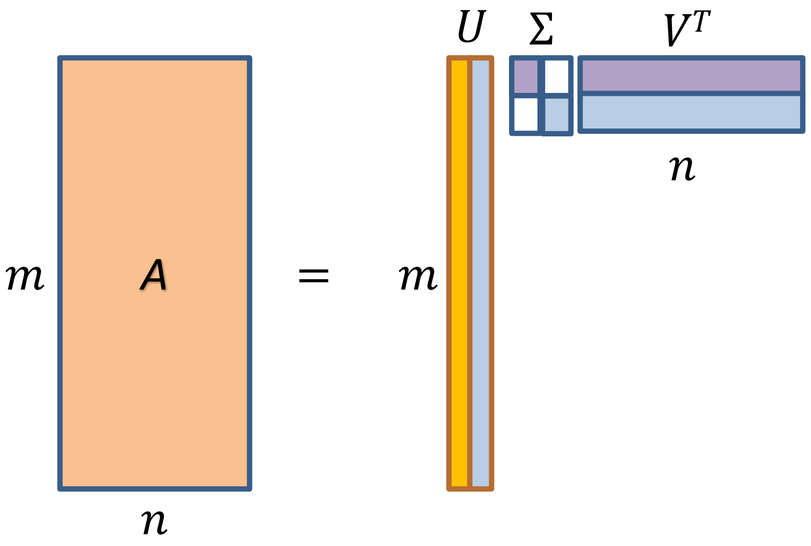
SVD: Definition

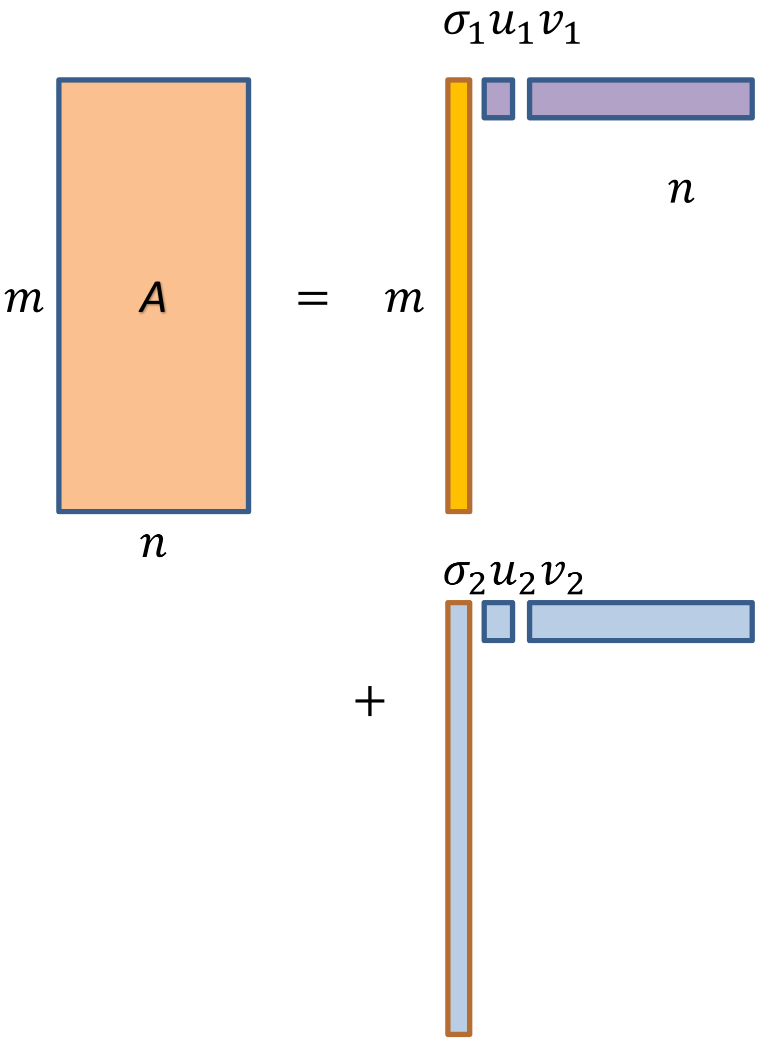
- $A_{[m \times n]} = U_{[m \times r]} \Sigma_{[r \times r]} V^T_{[n \times r]}$
- A: input data matrix
  - $m \times n$ matrix (e.g. m documents, n terms)
- U: left singular vectors
  - $m \times r$ matrix (m documents, r topics)
- Σ: singular values
  - $r \times r$ diagonal matrix (strength of each "topic")
  - $r = rank(A)$: rank of matrix A
- V: right singular vectors
  - $n \times r$ matrix (n terms, r topics)



- $A = U\Sigma V^T = \sum_i \sigma_i u_i \circ v_i^T$
  - σ_i: scalar
  - u_i: vector
  - v_i: vector

SVD: Definition





- $A = U\Sigma V^T = \sum_i \sigma_i u_i \circ v_i^T$
  - σ_i: scalar
  - u_i: vector
  - v_i: vector

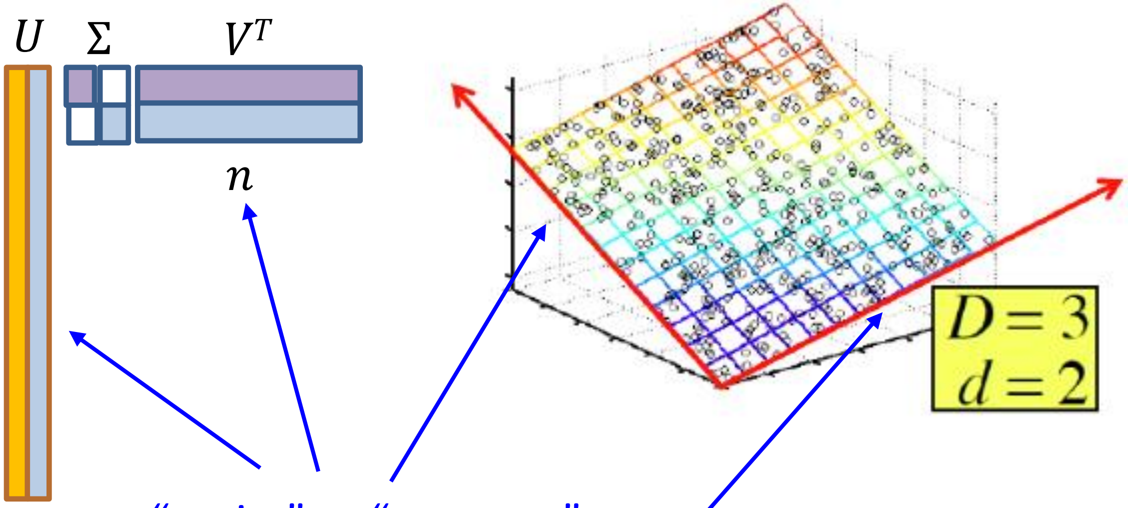
SVD Properties

- It is always possible to do SVD, i.e. decompose a matrix A into $A = U\Sigma V^T$, where
- U, Σ, V: unique
- U,V: column orthonormal
  - $U^T U = I, V^T V = I$ (I: identity matrix)
- Σ: diagonal
  - Entries (singular values) are non-negative,
  - Sorted in decreasing order ($\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$).

SVD: Example – Users-to-Movies

- $A = U\Sigma V^T$ - example: Users to Movies

$$\begin{array}{cc|cccc} & & \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight}\\ \hline \text{SciFi} & & 1 & 3 & 1 & 0\\ & & 3 & 4 & 3 & 1\\ & & 4 & 2 & 4 & 0\\ & m & 5 & 4 & 5 & 1\\ \text{Romance} & & 0 & 1 & 0 & 4\\ & & 0 & 0 & 0 & 3\\ & & 0 & 1 & 0 & 5\\ & & & n & & \end{array} = m$$



“topics” or “concepts”

aka. Latent dimensions

aka. Latent factors

SVD: Example – Users-to-Movies

- $A = U\Sigma V^T$ - example

$$m\ \underset{n}{\begin{array}{c|cccc} & \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \\ \hline \text{SciFi} & 1 & 3 & 1 & 0 \\ & 3 & 4 & 3 & 1 \\ & 4 & 2 & 4 & 0 \\ & 5 & 4 & 5 & 1 \\ \text{Romance} & 0 & 1 & 0 & 4 \\ & 0 & 0 & 0 & 3 \\ & 0 & 1 & 0 & 5 \end{array}} \overset{m}{=} \begin{bmatrix} \overset{\text{SciFi-concept}}{0.24} & \overset{\text{Romance-concept}}{0.02} & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{bmatrix} \text{X} \begin{bmatrix} 11.9 & 0 & 0 \\ 0 & 7.1 & 0 \\ 0 & 0 & 2.5 \end{bmatrix} \text{X}$$

$$\underset{n}{\begin{bmatrix} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0.37 & 0.83 & -0.37 & -0.17 \end{bmatrix}}$$

SVD: Example – Users-to-Movies

- $A = U\Sigma V^T$ - example

U is "user-to-concept" similarity matrix

$$
m\;
\begin{array}{c|cccc}
 & \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \\
\hline
\text{SciFi} & 1 & 3 & 1 & 0 \\
 & 3 & 4 & 3 & 1 \\
 & 4 & 2 & 4 & 0 \\
 & 5 & 4 & 5 & 1 \\
\text{Romance} & 0 & 1 & 0 & 4 \\
 & 0 & 0 & 0 & 3 \\
 & 0 & 1 & 0 & 5
\end{array}
\atop n
\;\overset{m}{=}\;
\begin{bmatrix}
\overset{\text{SciFi-concept}}{\boxed{0.24}} & \overset{\text{Romance-concept}}{0.02} & 0.69 \\
0.48 & -0.02 & 0.43 \\
0.49 & -0.08 & -0.52 \\
0.68 & -0.07 & -0.16 \\
0.06 & 0.57 & 0.06 \\
0.01 & 0.41 & -0.20 \\
0.07 & 0.70 & -0.01
\end{bmatrix}
\times
\begin{bmatrix}
11.9 & 0 & 0 \\
0 & 7.1 & 0 \\
0 & 0 & 2.5
\end{bmatrix}
\times
\underset{n}{
\begin{bmatrix}
0.59 & 0.54 & 0.59 & 0.05 \\
-0.10 & 0.12 & -0.10 & 0.98 \\
-0.37 & 0.83 & -0.37 & -0.17
\end{bmatrix}}
$$

SVD: Example – Users-to-Movies

- $A = U\Sigma V^T$ - example

SciFi-concept

“strength” of the SciFi-concept

$$\begin{array}{cc} & \begin{array}{cccc}\text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight}\end{array}\\ m\ \begin{array}{r}\text{SciFi}\left\{\vphantom{\begin{array}{c}1\\1\\1\\1\end{array}}\right.\\ \text{Romance}\left\{\vphantom{\begin{array}{c}1\\1\\1\end{array}}\right.\end{array} & \underset{n}{\begin{bmatrix}1&3&1&0\\3&4&3&1\\4&2&4&0\\5&4&5&1\\0&1&0&4\\0&0&0&3\\0&1&0&5\end{bmatrix}}\end{array} = m\begin{bmatrix}0.24&0.02&0.69\\0.48&-0.02&0.43\\0.49&-0.08&-0.52\\0.68&-0.07&-0.16\\0.06&0.57&0.06\\0.01&0.41&-0.20\\0.\,07&0.70&-0.01\end{bmatrix} \text{ X } \begin{bmatrix}11.9&0&0\\0&7.1&0\\0&0&2.5\end{bmatrix} \text{ X } \underset{n}{\begin{bmatrix}0.59&0.54&0.59&0.\,05\\-0.10&0.12&-0.10&0.98\\-0.\,37&0.83&-0.37&-0.17\end{bmatrix}}$$

SVD: Example – Users-to-Movies

- $A = U\Sigma V^T$ - example

***V* is “movie-to-concept” similarity matrix**

SciFi-concept

$$\begin{array}{c|cccc} & \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \\ \hline \text{SciFi} & 1 & 3 & 1 & 0 \\ & 3 & 4 & 3 & 1 \\ & 4 & 2 & 4 & 0 \\ & 5 & 4 & 5 & 1 \\ \text{Romance} & 0 & 1 & 0 & 4 \\ & 0 & 0 & 0 & 3 \\ & 0 & 1 & 0 & 5 \end{array}$$

$$m \underset{n}{\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix}} \overset{m}{=} \begin{bmatrix} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{bmatrix} \times \begin{bmatrix} 11.9 & 0 & 0 \\ 0 & 7.1 & 0 \\ 0 & 0 & 2.5 \end{bmatrix} \times \underset{n}{\begin{bmatrix} \boxed{0.59} & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0.37 & 0.83 & -0.37 & -0.17 \end{bmatrix}}$$

SciFi-concept

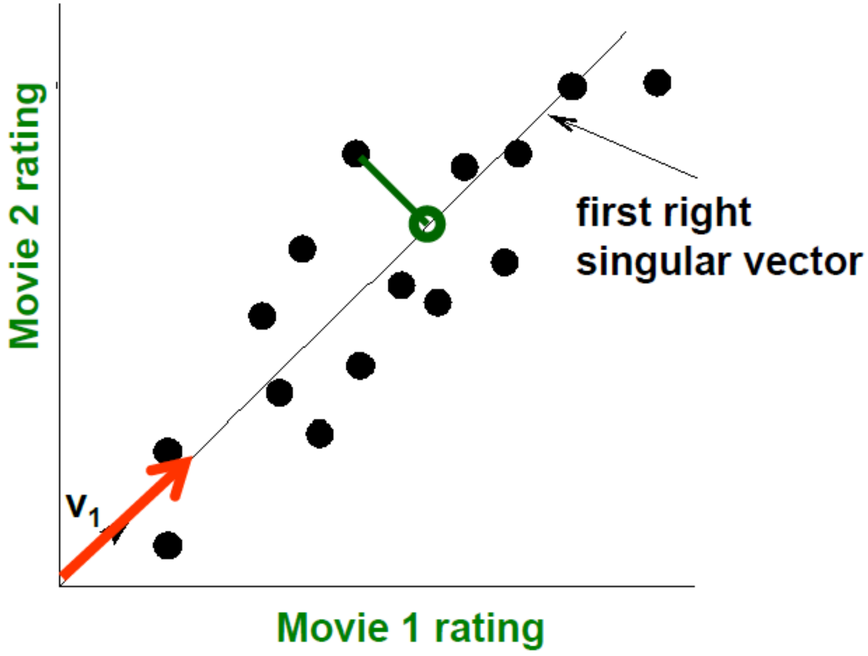
SVD: Interpretation #1

- “users”, “movies” and “concepts”
  - – U: user-to-concept similarity matrix
  - – V: movie-to-concept similarity matrix
  - – Σ: its diagonal elements
    - ‘strength’ of each concept

SVD: Interpretations #2

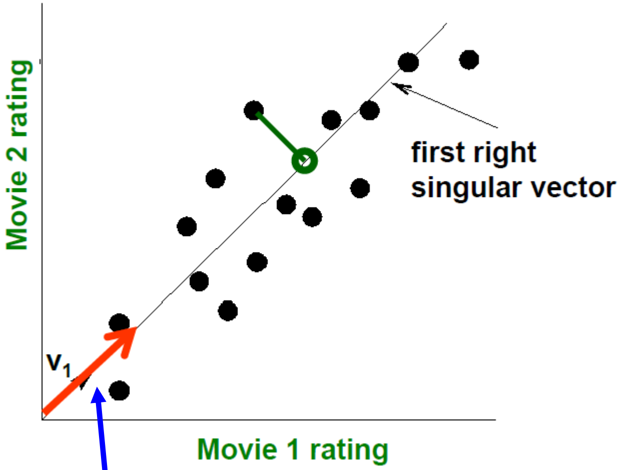
- SVD gives ‘best’ axis to project on
  - ‘best’ = minimal sum of squares of projection errors
- In other words,

  minimum reconstruction error



SVD: Interpretation #2

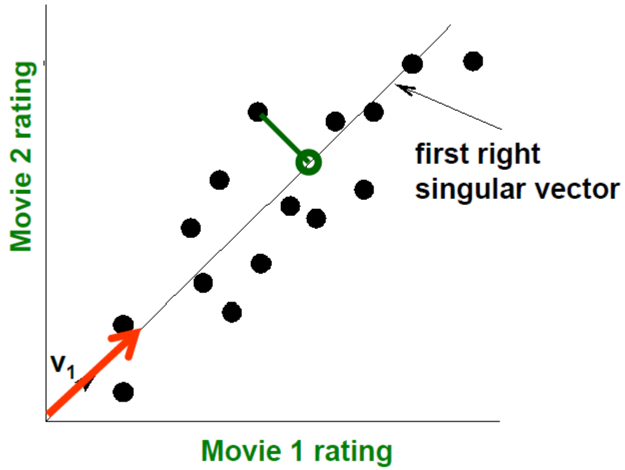
- $A = U\Sigma V^T$ - example
  - U: user-to-concept matrix
  - V: movie-to-concept matrix



$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} = \begin{bmatrix} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{bmatrix} \times \begin{bmatrix} 11.9 & 0 & 0 \\ 0 & 7.1 & 0 \\ 0 & 0 & 2.5 \end{bmatrix} \times \begin{bmatrix} \boxed{0.59 \quad 0.54 \quad 0.59 \quad 0.05} \\ -0.10 \quad 0.12 \quad -0.10 \quad 0.98 \\ -0.37 \quad 0.83 \quad -0.37 \quad -0.17 \end{bmatrix}$$

SVD: Interpretation #2

- $A = U\Sigma V^T$ - example



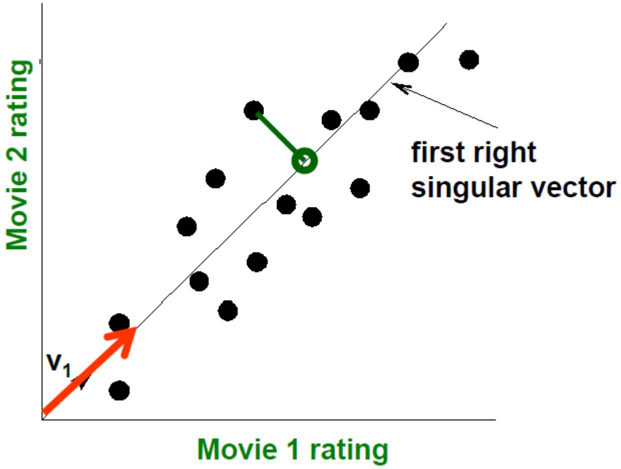
variance ("spread") on the v_1 axis

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} = \begin{bmatrix} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{bmatrix} \text{x} \begin{bmatrix} 11.9 & 0 & 0 \\ 0 & 7.1 & 0 \\ 0 & 0 & 2.5 \end{bmatrix} \text{x}$$

$$\begin{bmatrix} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0.37 & 0.83 & -0.37 & -0.17 \end{bmatrix}$$

SVD: Interpretation #2

- $A = U\Sigma V^T$ - example
  - $U\Sigma$: the coordinates of the points in the projection axis



Projection of users on the “Sci-Fi” axis

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \qquad \left[\begin{array}{|c|cc} \hline 2.86 & 0.24 & 8.21 \\ 5.71 & -0.24 & 5.12 \\ 5.83 & -0.95 & -6.19 \\ 8.09 & -0.83 & -1.90 \\ 0.71 & 6.78 & 0.71 \\ 0.12 & 4.88 & -2.38 \\ 0.83 & 8.33 & -0.12 \\ \hline \end{array}\right]$$

SVD: Interpretation #2

- Q: how exactly is dimension reduction done?

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} = \begin{bmatrix} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0. 07 & 0.70 & -0.01 \end{bmatrix} \text{ x } \begin{bmatrix} 11.9 & 0 & 0 \\ 0 & 7.1 & 0 \\ 0 & 0 & 2.5 \end{bmatrix} \text{ x }$$

$$\begin{bmatrix} 0.59 & 0.54 & 0.59 & 0. 05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0. 37 & 0.83 & -0.37 & -0.17 \end{bmatrix}$$

SVD: Interpretation #2

- Q: how exactly is dimension reduction done?
- A: Set smallest singular values to zero

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} = \begin{bmatrix} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{bmatrix} \text{x} \begin{bmatrix} 11.9 & 0 & 0 \\ 0 & 7.1 & 0 \\ 0 & 0 & \cancel{2.5} \end{bmatrix} \text{x}$$

$$\begin{bmatrix} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0.37 & 0.83 & -0.37 & -0.17 \end{bmatrix}$$

SVD: Interpretation #2

- Q: how exactly is dimension reduction done?
- A: Set smallest singular values to zero
  - Approximate original matrix by low-rank matrices

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \approx \begin{bmatrix} 0.24 & 0.02 & \xcancel{0.69} \\ 0.48 & -0.02 & \xcancel{0.48} \\ 0.49 & -0.08 & \xcancel{-0.52} \\ 0.68 & -0.07 & \xcancel{-0.16} \\ 0.06 & 0.57 & \xcancel{0.06} \\ 0.01 & 0.41 & \xcancel{-0.20} \\ 0.07 & 0.70 & \xcancel{-0.01} \end{bmatrix} \text{x} \begin{bmatrix} 11.9 & 0 & 0 \\ 0 & 7.1 & 0 \\ 0 & 0 & \xcancel{2.5} \end{bmatrix} \text{x}$$

$$\begin{bmatrix} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ \xcancel{-0.37} & \xcancel{0.83} & \xcancel{-0.37} & \xcancel{-0.17} \end{bmatrix}$$

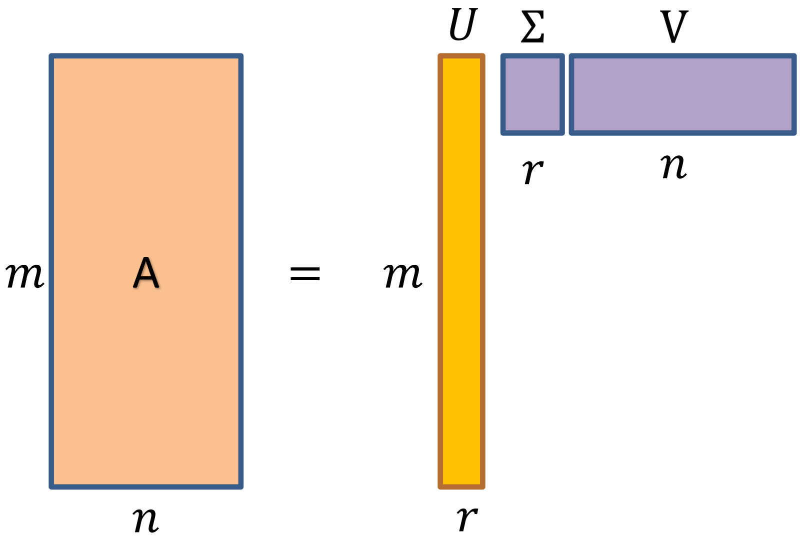
SVD: Interpretation #2

- Q: how exactly is dimension reduction done?
- A: Set smallest singular values to zero
  - Approximate original matrix by low-rank matrices

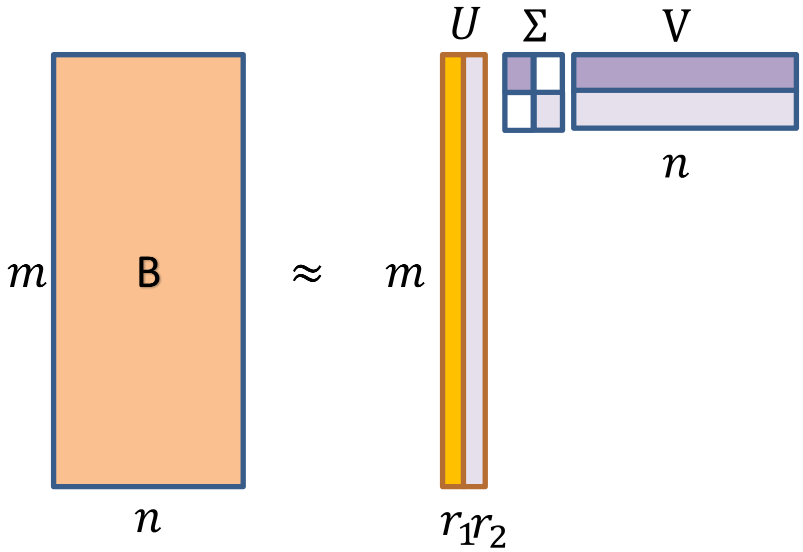
$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \approx \begin{bmatrix} 0.24 & 0.02 & \\ 0.48 & -0.02 & \\ 0.49 & -0.08 & \\ 0.68 & -0.07 & \\ 0.06 & 0.57 & \\ 0.01 & 0.41 & \\ 0. 07 & 0.70 & \end{bmatrix} \text{x} \begin{bmatrix} 11.9 & 0 & \\ 0 & 7.1 & \\ & & \end{bmatrix} \text{x}$$

$$\begin{bmatrix} 0.59 & 0.54 & 0.59 & 0. 05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ & & & \end{bmatrix}$$

SVD: Best Low Rank Approximation



A is a rank r matrix



B is the **best** rank r_1 approximation of matrix A

SVD: Best Low Rank Approximation

- **Theorem**: Let $A = U\Sigma V^T$ $(\mathrm{rank}(A) = r, \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r)$, and $B = USV^T$
  - S = diagonal $k \times k$ matrix where $s_i = \sigma_i$ $(i = 1 \ldots k)$ and $s_i = 0$ $(i > k)$
  - or equivalently, $B = \sum_{i=1}^{k} \sigma_i \, u_i \circ v_i^T$, is the best rank-$k$ approximation to A:
  - or equivalently, $B = \underset{rank(B) \leq k}{\operatorname{argmin}} \|A - B\|_F$
- Intuition (spectral decomposition)
  - $A = \sum_i \sigma_i u_i \circ v_i^T = \sigma_1 u_1 \circ v_1^T + \cdots + \sigma_r u_r \circ v_r^T$
    - $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$
  - Why setting small σ_i to 0 is the right thing to do?
    - Vectors u_i and v_i are unit length, so σ_i scales them.
    - Therefore, zeroing small σ_i introduces less error.

SVD: Interpretation #2

- Q: How many σ_i to keep?
- A: Rule-of-a thumb

Keep 80~90% "energy" $(= \sum_i {\sigma_i}^2)$

$$\begin{array}{cc} m & \begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \\ & n \end{array} = \sigma_1 u_1 \circ {v_1}^T + \sigma_2 u_2 \circ {v_2}^T + \cdots$$

Assume: $\sigma_1 \geq \sigma_2 \geq \cdots$

SVD: Complexity

- SVD for full matrix
  - $O(\min(nm^2, n^2m))$
- But
  - faster, if we only want to compute singular values
  - or if we only want first k singular vectors (thin-svd).
  - or if the matrix is sparse (sparse svd).
- Stable implementations
  - LAPACK, Matlab, PROPACK …
  - Available in most common languages

SVD: Conclusions so far

- SVD: $A = U\Sigma V^T$: unique
  - U: user-to-concept similarities
  - V: movie-to-concept similarities
  - Σ: strength to each concept
- Dimensionality reduction
  - Keep the few largest singular values (80-90% of "energy")
  - SVD: picks up linear correlations

SVD: Relationship to Eigen-decomposition

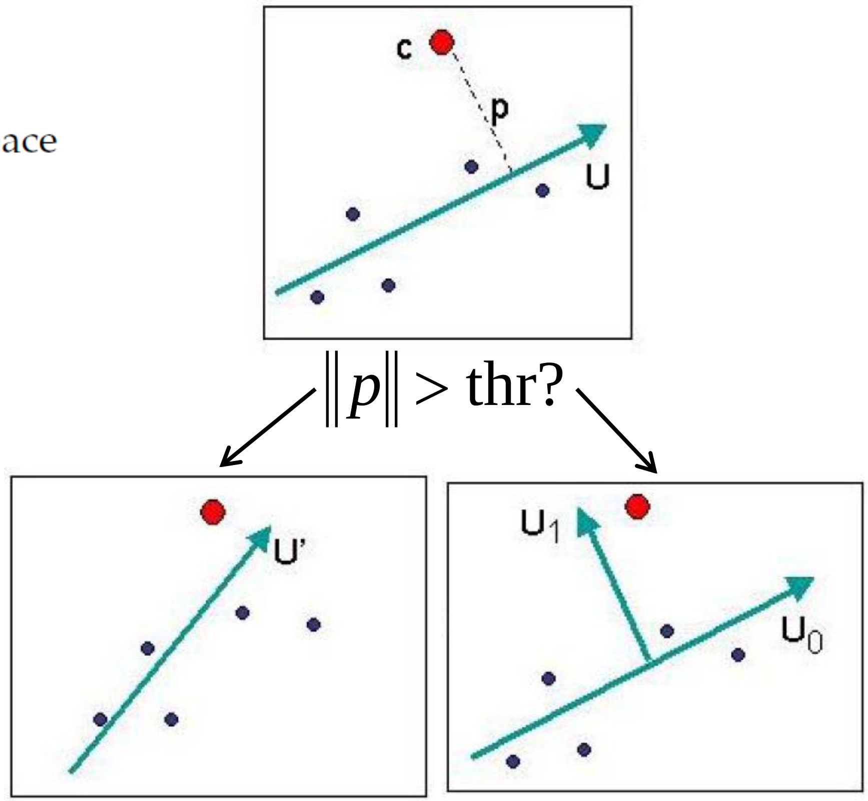
- SVD gives us
  - $A = U\Sigma V^T$
- Eigen-decomposition
  - $A = X\Lambda \mathrm{X}^{\mathrm{T}}$
    - A is symmetric
    - U, V, X are orthonormal ($U^T U = I$)
    - Λ, Σ are diagonal
- Equivalence
  - $AA^T = U\Sigma V^T (U\Sigma V^T)^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T = X\Lambda X^T$
  - $A^T A = V\Sigma^T U^T (U\Sigma V^T) = V\Sigma^T \Sigma V^T = Y\Lambda Y^T$
  - This shows how to use eigen-decomposition to compute SVD
  - And also, $\lambda_i = \sigma_i^2$

Online SVD (Brand, 2003)

- Challenges: storage and computation
- Idea: an incremental algorithm computes the principal eigenvectors of a matrix without storing the entire matrix in memory

Online SVD (Brand, 2003)

```
Existing rank-$r$ PCA
    $A = U\Sigma V^T$
A new sample $c$ arrives, project it onto eigenspace
    $m = U^T c$
Compute the orthogonal component
    $p = c - Um$
if $\|p\| < thr$ then
    Incorporate the new sample by rotating $U$
        $U = UR_u, \quad V = VR_v$
else
    increase a rank
        $U' = [U; m]R_u, \quad V' = VR_v$
end if
Rotation by re-diagonalizing the matrix
    $\begin{pmatrix} \text{diag}\ (S) & m \\ 0 & \|p\| \end{pmatrix} \longrightarrow [R_u, R_v]$
```



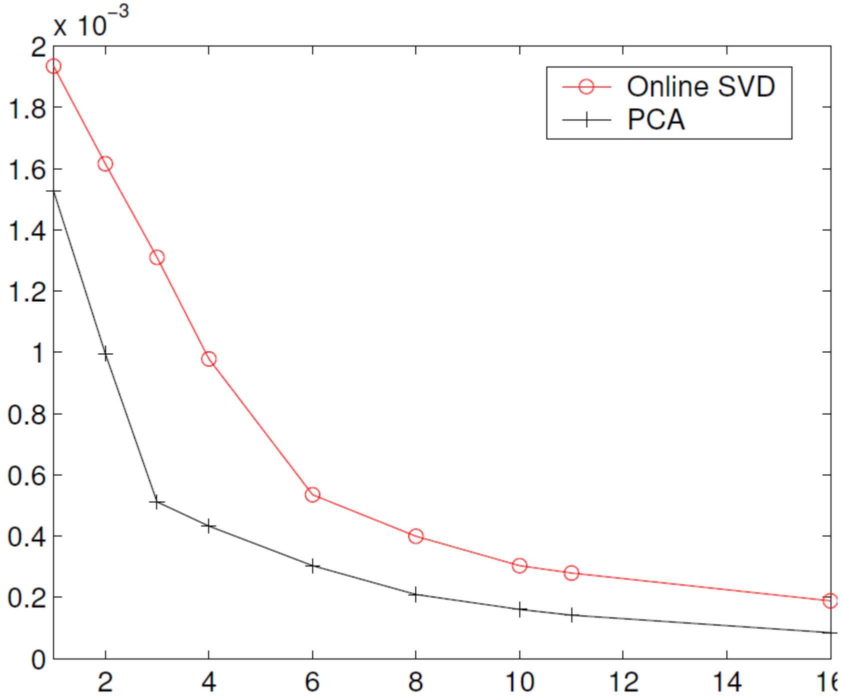
Online SVD (Brand, 2003)

- Complexity

$$O(r^2)$$

- Store
  - U, S, V

- The online SVD has more error than the PCA



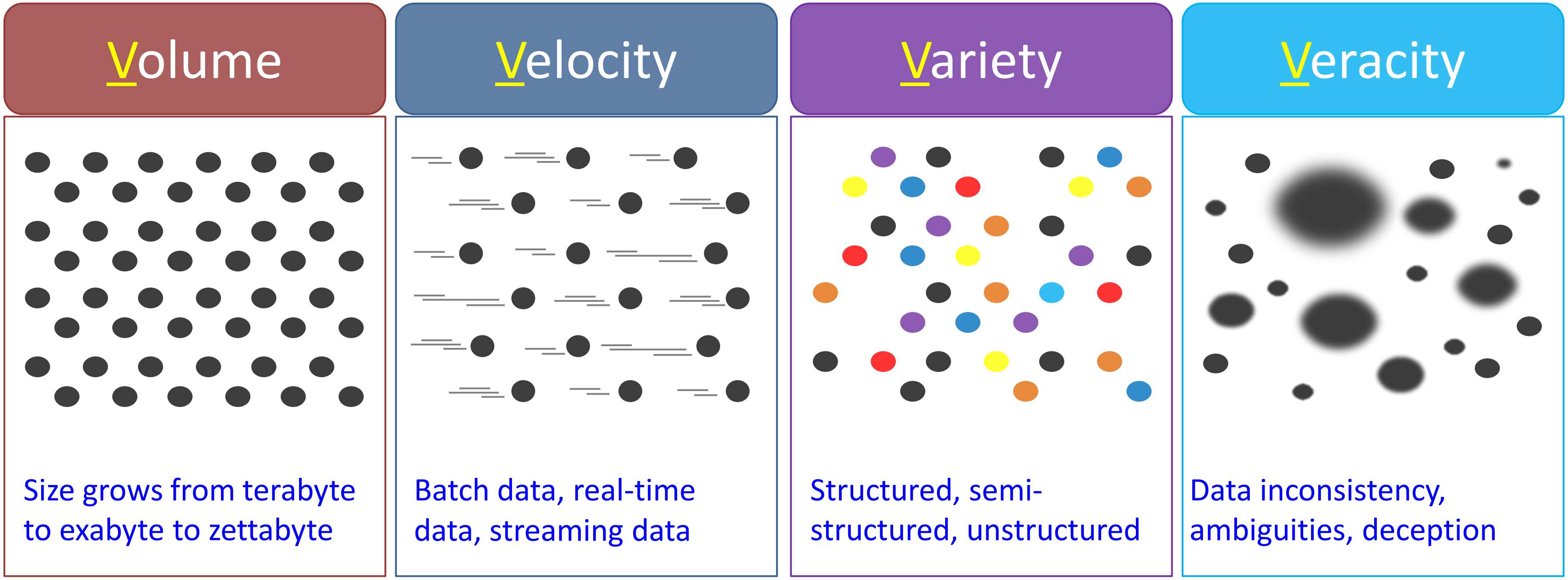
Online Unsupervised Learning

- Unsupervised learning: minimizing the reconstruction errors
- Online: rank-one update
- Pros and Cons
  - 😆 Simple to implement
  - 😆 Heuristic, but intuitively work
  - 😭 Lack of theoretical guarantee
  - 😭 Relative poor performance

Outline

- Introduction (60 min.)
  - Big data and big data analytics (30 min.)
  - Online learning and its applications (30 min.)
- Online Learning Algorithms (60 min.)
  - Perceptron (10 min.)
  - Online non-sparse learning (10 min.)
  - Online sparse learning (20 min.)
  - Online unsupervised learning (20. min.)
- Discussions + Q & A (5 min.)

Discussions and Open Issues



Volume

Size grows from terabyte to exabyte to zettabyte

Velocity

Batch data, real-time data, streaming data

Variety

Structured, semi-structured, unstructured

Veracity

Data inconsistency, ambiguities, deception

How to learn from Big Data to tackle the 4V's characteristics?

Discussions and Open Issues

- Data issues
  - High-dimensionality
  - Sparsity
  - Structure
  - Noise and incomplete data
  - Concept drift
  - Domain adaption
  - Background knowledge incorporation
- Platform issues
  - Parallel computing
  - Distributed computing
- User interaction
  - Interactive OL vs. Passive OL
  - Crowdsourcing

Discussions and Open Issues

- Applications
  - Social network and social media
  - Speech recognition and identification (e.g., Siri)
  - Financial engineering
  - Medical and healthcare informatics
  - Science and research: human genome decoding, particle discoveries, astronomy
  - etc.

Conclusion

- Introduction of Big Data and the challenges and opportunities
- Introduction of online learning and its possible applications
- Survey of classical and state-of-the-art online learning techniques for
  - Non-sparse learning models
  - Sparse learning models
  - Unsupervised learning models

One-slide Takeaway

- Online learning is a promising tool for big data analytics
- Many challenges exist
  - Real-world scenarios: concept drifting, sparse data, high-dimensional, uncertain/imprecision data, etc.
  - More advance online learning algorithms: faster convergence rate, less memory cost, etc.
  - Parallel implementation or running on distributing platforms

Toolbox: http://appsrv.cse.cuhk.edu.hk/~hqyang/doku.php?id=software

Other Toolboxes

- MOA: Massive Online Analysis
  - http://moa.cms.waikato.ac.nz/
- Vowpal Wabbit
  - https://github.com/JohnLangford/vowpal_wabbit/wiki

Q & A

- If you have any problems, please send emails to hqyang@cse.cuhk.edu.hk!

References

- M. Brand. Fast online svd revisions for lightweight recommender systems. In SDM, 2003.
- N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order perceptron algorithm. SIAM J. Comput., 34(3):640–668, 2005.
- K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. Journal of Machine Learning Research, 7:551–585, 2006.
- K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. In NIPS, pages 414–422, 2009.
- K. Crammer and D. D. Lee. Learning via gaussian herding. In NIPS, pages 451–459, 2010.
- K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. Journal of Machine Learning Research, 3:951–991, 2003.
- M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In ICML, pages 264–271, 2008.
- J. C. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. Journal of Machine Learning Research, 10:2899–2934, 2009.
- C. Gentile. A new approximate maximal margin classification algorithm. Journal of Machine Learning Research, 2:213–242, 2001.
- M. D. Hoffman, D. M. Blei, and F. R. Bach. Online learning for latent dirichlet allocation. In NIPS, pages 856–864, 2010.

References

- S. C. H. Hoi, J. Wang, and P. Zhao. Exact soft confidence-weighted learning. In ICML, 2012.
- J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. Journal of Machine Learning Research, 10:777–801, 2009.
- Y. Li and P. M. Long. The relaxed online maximum margin algorithm. Machine Learning, 46(1-3):361–387, 2002.
- P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal., 16(6):964–979, 1979.
- J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In ICML, page 87, 2009.
- Y. Nesterov. Gradient methods for minimizing composite objective function. CORE Discussion Paper 2007/76, Catholic University of Louvain, Center for Operations Research and Econometrics, 2007.
- F. Orabona and K. Crammer. New adaptive algorithms for online classification. In NIPS, pages 1840–1848, 2010.
- F. Rosenblatt. The Perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65:386–408, 1958.

References

- C. Wang, J. W. Paisley, and D. M. Blei. Online variational inference for the hierarchical dirichlet process. Journal of Machine Learning Research - Proceedings Track, 15:752–760, 2011.
- M. K. Warmuth and D. Kuzmin. Randomized pca algorithms with regret bounds that are logarithmic in the dimension. In NIPS, pages 1481–1488, 2006.
- S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. IEEE Transactions on Signal Processing, 57(7):2479–2493, 2009.
- L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. Journal of Machine Learning Research, 11:2543–2596, 2010.
- H. Yang, M. R. Lyu, and I. King. Efficient online learning for multi-task feature selection. ACM Transactions on Knowledge Discovery from Data, 2013.
- H. Yang, Z. Xu, I. King, and M. R. Lyu. Online learning for group lasso. In ICML, pages 1191–1198, Haifa, Israel, 2010.
- L. Yang, R. Jin, and J. Ye. Online learning by ellipsoid method. In ICML, page 145, 2009.
- P. Zhao, S. C. H. Hoi, and R. Jin. Duol: A double updating approach for online learning. In NIPS, pages 2259–2267, 2009.
- M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In ICML, pages 928–936, 2003.